SφT

NOV 1983
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

NASA SP-221(06)

THE NASTRAN® THEORETICAL MANUAL

(NASA-SP-221(06)) THE NASTRAN THEORETICAL MANUAL (National Aeronautics and Space Administration) 944 p HC A99/MF A01 N84-18677
CSCL 20K Unclas
H1/39 12458

MAR 1984
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

January 1981

OCT 1983
RECEIVED
NASA STI FACILITY
ACQ BRANCH

Scientific and Technical Information Office
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C.

INTRODUCTION

The documentation for the NASTRAN computer program consists of four manuals: the Theoretical Manual, the User's Manual, the Programmer's Manual and the Demonstration Problem Manual. Since an effort has been made to avoid duplication of material, a brief statement of the content of the other three manuals will serve as a useful point of departure in introducing the Theoretical Manual.

The intent of the User's Manual is that it provide all of the information needed to solve problems with NASTRAN. Users should find it to be both instructional and encyclopedic. It includes instruction in structural modeling techniques, instruction in input preparation and information to assist the interpretation of output. It contains descriptions of all input data cards, restart procedures and diagnostic messages. It is hoped that it can serve as a self-help instruction book.

The intent of the Programmer's Manual is that it provide a complete description of the program code, including the mathematical equations that are implemented in the Functional Modules. It describes the Executive System and the coding practices that have been employed. It contains the information that is required for maintenance and modification of the program.

The intent of the Demonstration Problem Manual is to illustrate the formulation of the types of problems that can be solved with NASTRAN and to show that the results obtained are valid. Generally, this manual discusses the nature of the problem, the underlying theory, the specific geometric and physical input quantities, and the comparison of theoretical and NASTRAN results. At least one problem for each of the rigid formats and nearly all of the elements is provided.

One of the roles that has been assigned to the Theoretical Manual is that of a commentary on the program. It is, first of all, intended to be an introduction to NASTRAN for all interested persons, including those who will go on to use the program and those whose interests are less direct. For this purpose, the structure and the problem solving capabilities of the program are described in a narrative style. The manual's most important function, however, is to present developments of the analytical and numerical procedures that underlie the program.

The selection of material for the Theoretical Manual has not been an easy task because not everyone has the same concept of what the word "theory" means when it is applied to a computer program. For some, theory is restricted to include only the formulation of the equations that will be solved; for others, theory also includes the development of the procedures, or algorithms, that will be used in the solution; still others regard the organization of the program and the flow of data through the computer as important theoretical topics.

A broad view concerning the selection of material has been adopted, and the reader will find that all of the above aspects of the program are treated. Some structural analysts may be surprised at the emphasis on program organization and data processing, particularly in the early sections of the manual. These subjects are emphasized because they are vitally important to the success of a large computer program and should not be taken for granted.

In regard to the more mathematical subjects, such as the derivation of the equations for structural elements and the development of eigenvalue extraction procedures, the reader will find that the level of sophistication is geared to the difficulty of the subject matter. Thus, it is assumed that a reader with an interest in an advanced topic (such as shell elements) will have the necessary theoretical background. In most cases the derivations are intended to be complete and rigorous. For a few of the structural elements, the reader is referred to the Programmer's Manual for the detailed expression of matrix coefficients that are regarded as too combersome to have general interest.

The Theoretical Manual is divided into seventeen major sections and numerous subsections. Section 1 deals with some of the organizational aspects of NASTRAN and Section 2 with utility matrix routines. Sections 3, 4, 5 and 7 deal with static structural analysis. It will be noted that no material has been included in Section 6, which is reserved for topics to be defined in the future. Section 8 treats heat transfer. Sections 9 through 12 deal with dynamic structural analysis. Sections 13 through 15 deal with miscellaneous topics, including computer graphics, special structural modeling techniques and error analysis. Section 16 deals with the interaction between structures and fluids. Section 17 deals with aeroelastic analysis.

The style of the Theoretical Manual, like that of the other three manuals, has been designed to accommodate future additions and modifications. Each major subsection stands alone with its own page numbers, equation numbers and figure numbers, so that changes can be made without significant disruption.

CONTRIBUTORS TO THE THEORETICAL MANUAL

Section(s)	Author(s)
1.1	R. MacNeal
1.2	P. Peabody K. Redner
1.3	R. MacNeal
2.1	K. Redner S. Wall
2.2 and 2.3	C. McCormick
3.1 thru 3.3	R. MacNeal
3.4	C. McCormick R. MacNeal
3.5 thru 3.5.5	R. MacNeal
3.5.6	P. Pamidi
3.6 and 3.7	R. MacNeal
3.8	D. Herting
4.1 thru 4.4	R. MacNeal
4.5	R. MacNeal R. Harder
4.6	E. Field D. Herting
4.7	D. Herting
5.1 thru 5.5	R. MacNeal
5.5.1 and 5.5.2	T. Butler
5.5.3	R. MacNeal
5.5.4	T. Butler
5.6	R. MacNeal
5.7	R. MacNeal R. Gellatly
5.8.1 thru 5.8.4	R. MacNeal
5.8.5	H. Adelman J. Walz
5.8.6 and 5.8.7	R. Narayanaswami
5.9	R. MacNeal
5.10	J. Batt
5.11	S. Jordan
5.12	R. Harder
5.13	E. Field S. Johnson
5.14	R. Narayanaswami
7.1 and 7.2	R. MacNeal
7.3	D. Herting
7.4	R. MacNeal
7.5	E. Field S. Johnson
7.6	R. Narayanaswami
8.1	R. MacNeal
8.2 and 8.3	R. Harder
8.4	R. MacNeal D. Herting
9.1 thru 9.2.1	R. MacNeal
9.2.2	P. Pamidi
9.2.3 thru 9.4	R. MacNeal
10.1	R. MacNeal
10.2	T. Clark
10.3 and 10.4	R. MacNeal
10.5	R. MacNeal R. Harder
10.6	M. Newman
11.1 and 11.2	R. MacNeal D. Herting
11.3	R. MacNeal
11.4.1	R. MacNeal D. Herting
11.4.2	P. Pamidi
11.5	R. MacNeal
11.6	W. Rodden R. Harder E. Bellinger
12.1 thru 12.3	R. MacNeal
13.1	B. Bata C. Hennrich
13.2 and 13.3	C. Hennrich
14.1 and 14.2	R. MacNeal
15.1 and 15.2	C. McCormick
15.3	R. MacNeal
15.4	D. Herting L. Kuusinen
16.1 and 16.2	D. Herting R. MacNeal
17.1 thru 17.3	R. MacNeal R. Harder
17.4 thru 17.6	W. Rodden R. Harder E. Bellinger

TABLE OF CONTENTS

Section | Page No.

1. PROGRAM ORGANIZATION

- 1.1 OVERVIEW OF THE PROGRAM 1.1-1
- 1.2 THE NASTRAN EXECUTIVE SYSTEM 1.2-1
  - 1.2.1 Introduction 1.2-1
  - 1.2.2 Executive Operations During the Preface 1.2-4
  - 1.2.3 Executive Operations During Problem Solution 1.2-7
- 1.3 USER CONTROL OF PROBLEM EXECUTION 1.3-1

2. MATRIX OPERATIONS

- 2.1 ELEMENTARY OPERATIONS 2.1-1
  - 2.1.1 Introduction 2.1-1
  - 2.1.2 Matrix Multiplication 2.1-3
  - 2.1.3 Matrix Addition 2.1-4a
  - 2.1.4 Matrix Transposition 2.1-5
  - 2.1.5 Matrix Partitioning and Merging 2.1-6
- 2.2 TRIANGULAR DECOMPOSITION 2.2-1
  - 2.2.1 Triangular Decomposition of Symmetrical Matrices 2.2-1
  - 2.2.2 Triangular Decomposition of Unsymmetrical Matrices 2.2-6
- 2.3 SOLUTION OF [A]{x} = {b} 2.3-1

3. STATIC ANALYSIS BY THE DISPLACEMENT METHOD

- 3.1 INTRODUCTION 3.1-1
- 3.2 GENERAL PROBLEM FLOW 3.2-1
- 3.3 NOTATION SYSTEM 3.3-1
- 3.4 PROBLEM FORMULATION 3.4-1
  - 3.4.1 Structural Modeling 3.4-1
  - 3.4.2 Input Data Analysis 3.4-4
  - 3.4.3 Geometry Processor and Structure Plotter 3.4-5
  - 3.4.4 Assembly of Structural Matrices 3.4-6
- 3.5 CONSTRAINTS AND PARTITIONING 3.5-1
  - 3.5.1 Multipoint Constraints and Rigid Elements 3.5-1
  - 3.5.2 Single Point Constraints 3.5-3

5

TABLE OF CONTENTS (Continued)

Section Page No.

3.5.3 Partitioning 3.5-4

3.5.4 The Guyan Reduction 3.5-6

3.5.5 Special Provisions for Free Bodies 3.5-7

3.5.6 Rigid Elements 3.5-10

3.6 STATIC LOADS 3.6-1

3.6.1 Generation of Loads 3.6-1

3.6.2 Reduction of Load Vectors to Final Form and Solution for Displacements 3.6-4

3.6.3 Inertia Relief 3.6-6

3.7 DATA RECOVERY 3.7-1

3.7.1 Recovery of Displacements 3.7-1

3.7.2 Recovery of Stress Data 3.7-2

3.7.3 Grid Point Force Balance and Element Strain Energy Distribution 3.7-4

3.7.4 Overall Equilibrium Check 3.7-5

3.8 PIECEWISE LINEAR ANALYSIS 3.8-1

3.8.1 Limitations and Available Options 3.8-1

3.8.2 Overall Solution Logic 3.8-2

3.8.3 Piecewise Linear Stress-Strain Functions 3.8-3

3.8.4 Element Algorithms for Piecewise Linear Analysis 3.8-4

4. MISCELLANEOUS GENERAL PROVISIONS FOR STRUCTURAL ANALYSIS

4.1 AXISYMMETRIC STRUCTURES 4.1-1

4.1.1 Axisymmetric Element Library 4.1-1

4.1.2 Coordinate Systems 4.1-1

4.1.3 Harmonic Coefficients and Degrees of Freedom 4.1-2

4.1.4 Application of Loads 4.1-3

4.1.5 Differential Stiffness 4.1-4

4.1.6 Hydroelastic Capability 4.1-4

4.2 STRUCTURAL MATERIAL PROPERTIES 4.2-1

4.3 MANUAL SINGLE-STAGE SUBSTRUCTURE ANALYSIS 4.3-1

4.4 FULLY STRESSED DESIGN OPTIMIZATION 4.4-1

TABLE OF CONTENTS (Continued)

Section Page No.

4.5 CYCLIC SYMMETRY 4.5-1
    4.5.1 Theory for Rotational Symmetry 4.5-2
    4.5.2 Theory for Dihedral Symmetry 4.5-8
    4.5.3 Advantages 4.5-11
4.6 AUTOMATED MULTI-STAGE SUBSTRUCTURE ANALYSIS 4.6-1
    4.6.1 Static Matrix Reduction 4.6-1
    4.6.2 Modal Matrix Reduction 4.6-4
    4.6.3 Substructure Combinations 4.6-8
    4.6.4 Solution and Recovery 4.6-16
4.7 COMPONENT MODE SYNTHESIS 4.7-1
    4.7.1 Real Modal Reduction 4.7-3
    4.7.2 Complex Modal Reduction 4.7-10
    4.7.3 Solution Data Recovery 4.7-12
    4.7.4 User Input Modes 4.7-16

5. STRUCTURAL ELEMENTS FOR THE DISPLACEMENT METHOD

5.1 INTRODUCTION 5.1-1
5.2 RODS AND BEAMS 5.2-1
    5.2.1 The BAR Element 5.2-2
    5.2.2 ROD Elements 5.2-5
5.3 SHEAR PANELS AND TWIST PANELS 5.3-1
5.4 CONSTRAINTS 5.4-1
5.5 TREATMENT OF INERTIA PROPERTIES 5.5-1
    5.5.1 Grid Point Mass 5.5-1
    5.5.2 Rigid Body Inertia at a Grid Point 5.5-2
    5.5.3 Inertia Properties of Structural Elements 5.5-8
    5.5.4 Grid Point Weight Generator 5.5-12
5.6 SCALAR ELEMENTS 5.6-1
5.7 THE GENERAL ELEMENT 5.7-1
5.8 PLATES 5.8-1
    5.8.1 The Membrane Triangle 5.8-2
    5.8.2 The Basic Bending Triangle 5.8-7
    5.8.3 Composite Plate Elements 5.8-21
    5.8.4 Inertia Properties of Plate Elements 5.8-28c

TABLE OF CONTENTS (Continued)

Section Page No.

5.8.5 The Isoparametric Quadrilateral Membrane Element, QDMEM1 ..... 5.8-33

5.8.6 The Linear Strain Triangular Membrane Element, TRIM6 .... 5.8-43

5.8.7 Higher Order Triangular Plate Bending Element, TRPLT1 .... 5.8-55

5.9 THE CONICAL SHELL ELEMENT 5.9-1

5.9.1 Coordinate Notation 5.9-1

5.9.2 Harmonic Dependence on Azimuth Position 5.9-1

5.9.3 Cases to be Treated 5.9-2

5.9.4 General Plan for Deriving the Stiffness Matrices 5.9-3

5.9.5 Stiffness Matrix for Finite Shear Flexibility, n > 0 .... 5.9-6

5.9.6 Stiffness Matrix for Finite Transverse Shear Flexibility, n = 0 .... 5.9-21

5.9.7 Stiffness Matrix for Zero Transverse Shear Flexibility, n > 0 .... 5.9-21

5.9.8 Stiffness Matrix for Zero Transferse Shear Flexibility, n = 0 .... 5.9-22

5.9.9 Static Loads 5.9-22

5.9.10 Data Reduction 5.9-32

5.10 THE DOUBLY CURVED SHELL ELEMENT 5.10-1

5.10.1 Introduction 5.10-1

5.10.2 Coordinate System Definition 5.10-3

5.10.3 Displacement Functions 5.10-6

5.10.4 Potential Energy 5.10-11

5.10.5 Element Stiffness Matrix 5.10-15

5.10.6 Load Vectors 5.10-17

5.10.7 Stress Recovery 5.10-23

5.10.8 Kinetic Energy and Mass Matrix 5.10-28

5.11 SOLID OF REVOLUTION ELEMENTS 5.11-1

5.11.1 General Development for the Axisymmetric Solid Elements 5.11-2

5.11.2 Deriving the Stiffness Matrices 5.11-4

5.11.3 Material Definitions 5.11-8

5.11.4 Triangular Ring Elements 5.11-9

5.11.5 Trapezoidal Ring Elements 5.11-14

5.11.6 Development of Non-Axisymmetric Load Vectors 5.11-16

5.11.7 Load Vectors for the TRIARG and TRAPRG Elements 5.11-18

5.11.8 Stress and Displacement Recovery 5.11-25

8

TABLE OF CONTENTS (Continued)

Section Page No.

5.12 CONSTANT STRAIN SOLID ELEMENTS 5.12-1
5.12.1 Introduction 5.12-1
5.12.2 Displacement Functions for the Tetrahedron 5.12-2
5.12.3 Strain, Stress and Potential Energy 5.12-2
5.12.4 Matrices in Basic Coordinates 5.12-5
5.12.5 Wedge and Hexahedron Elements 5.12-5
5.12.6 Stress Recovery 5.12-6
5.13 ISOPARAMETRIC HEXAHEDRON SOLID ELEMENTS 5.13-1
5.13.1 Displacement Functions for the Isoparametric Element 5.13-2
5.13.2 Strain-Displacement Relationship 5.13-3
5.13.3 Stress-Strain Relations 5.13-5
5.13.4 Stiffness, Mass and Load Matrices 5.13-6
5.13.5 Numerical Integration 5.13-8
5.13.6 Transformation from Basic to Global Coordinates 5.13-8
5.13.7 Stress Recovery 5.13-9
5.14 THE TRIANGULAR THIN SHALLOW SHELL ELEMENT 5.14-1
5.14.1 Introduction 5.14-1
5.14.2 Element Geometry 5.14-1
5.14.3 Displacement Field 5.14-1
5.14.4 Derivation of the Stiffness Matrix 5.14-2
5.14.5 Equivalent Thermal Load Vector 5.14-7

7. DIFFERENTIAL STIFFNESS AND BUCKLING

7.1 INTRODUCTION 7.1-1
7.2 RODS, BEAMS, AND SHEAR PANELS 7.2-1
7.2.1 Extensional Rods 7.2-1
7.2.2 Quadrilateral Shear Penals 7.2-2
7.2.3 Prismatic Beams 7.2-3
7.3 PLATES 7.3-1
7.3.1 The Energy of Differential Stiffness 7.3-1
7.3.2 Differential Stiffness for the Basic Bending Triangle 7.3-6

TABLE OF CONTENTS (Continued)

Section Page No.

7.3.3 The Membrane Triangle (TRMEM) and Quadrilateral (QDMEM) 7.3-11

7.3.4 The Triangular Composite Elements (TRIA1 and TRIA2) 7.3-12

7.3.5 The Quadrilateral Composite Elements (QUAD1 and QUAD2) 7.3-13

7.4 THE CONICAL SHELL ELEMENT 7.4-1

7.4.1 General Method 7.4-1

7.4.2 Evaluation of $[K_n^{qd}]$ 7.4-5

7.5 ISOPARAMETRIC HEXAHEDRON THREE-DIMENSIONAL SOLID ELEMENTS 7.5-1

7.6 THE TRIANGULAR THIN SHALLOW SHELL ELEMENT 7.6-1

8. HEAT TRANSFER ANALYSIS

8.1 GENERAL FEATURES 8.1-1

8.2 VOLUME HEAT CONDUCTION ELEMENTS 8.2-1

8.2.1 Constant Gradient Heat Conduction Elements 8.2-3

8.2.2 Three-Dimensional Isoparametric Solid Element Heat Transfer Matrices 8.2-8

8.3 SURFACE HEAT TRANSFER 8.3-1

8.3.1 Prescribed Heat Flux 8.3-1

8.3.2 Convective Heat Flux 8.3-2

8.3.3 Radiation from a Distant Source 8.3-3

8.3.4 Radiation Exchange Between Surfaces 8.3-4

8.4 METHODS OF SOLUTION 8.4-1

8.4.1 Nonlinear Steady-State Analysis 8.4-1

8.4.2 Transient Analysis 8.4-7

9. ORGANIZATION OF DYNAMIC ANALYSIS

9.1 GENERAL PROBLEM FLOW IN DYNAMIC ANALYSIS 9.1-1

9.2 EIGENVALUE ANALYSIS 9.2-1

9.2.1 Real Eigenvalue Analysis 9.2-1

9.2.2 The APPEND Feature for Real Eigenvalue Analysis 9.2-6

9.2.3 Complex Eigenvalue Analysis 9.2-7

9.3 ASSEMBLY OF DYNAMIC MATRICES 9.3-1

9.3.1 Notation System 9.3-1

TABLE OF CONTENTS (Continued)

Section Page No.

9.3.2 Extra Points and Transfer Functions 9.3-3

9.3.3 Direct Dynamic Matrix Assembly 9.3-6

9.3.4 Modal Dynamic Matrix Assembly 9.3-8

9.4 DYNAMIC DATA RECOVERY 9.4-1

10. EIGENVALUE EXTRACTION METHODS

10.1 INTRODUCTION 10.1-1

10.2 THE TRIDIAGONAL METHOD 10.2-1

10.2.1 Introduction 10.2-1

10.2.2 Tridiagonalization by the Givens Method 10.2-1

10.2.3 Extraction of the Eigenvalues of a Tridiagonal Matrix 10.2-5

10.2.4 Computation of Eigenvectors 10.2-10

10.3 THE DETERMINANT METHOD 10.3-1

10.3.1 Fundamentals of the Determinant Method 10.3-1

10.3.2 Iteration Algorithm 10.3-2

10.3.3 Scaling 10.3-3

10.3.4 Sweeping of Previously Extracted Eigenvalues 10.3-4

10.3.5 Search Procedures 10.3-5

10.3.6 Convergence Criteria 10.3-7

10.3.7 Test for Closeness to Starting Point 10.3-9

10.3.8 Recovery of Eigenvectors 10.3-10

10.4 THE INVERSE POWER METHOD WITH SHIFTS 10.4-1

10.4.1 Introduction 10.4-1

10.4.2 Theory for Real Eigenvalue Analysis 10.4-3

10.4.3 Summary of Procedures for Real Eigenvalue Analysis 10.4-18

10.4.4 Theory for Complex Eigenvalue Analysis 10.4-24

10.4.5 Summary of Procedures for Complex Eigenvalue Analysis 10.4-37

10.5 THE UPPER HESSENBERG METHOD 10.5-1

10.5.1 Reduction to Canonical Form 10.5-1

10.5.2 Reduction to Upper Hessenberg Form 10.5-2

10.5.3 The QR Iteration 10.5-2

TABLE OF CONTENTS (Continued)

Section Page No.

10.5.4 Convergence Criteria 10.5-3
10.5.5 Shifting 10.5-3
10.5.6 Deflation 10.5-4
10.5.7 Eigenvectors 10.5-4
10.6 THE TRIDIAGONAL REDUCTION (FEER) METHOD 10.6-1
10.6.1 Introduction 10.6-1
10.6.2 Theory for Real Eigenvalue Analysis 10.6-3
10.6.3 Summary of Computational Procedures and Flow Charts - Real Eigenvalue Analysis 10.6-25
10.6.4 Theory for Complex Eigenvalue Analysis 10.6-32
10.6.5 Summary of Computational Procedures and Flow Charts - Complex Eigenvalue Analysis 10.6-44

11. TRANSIENT ANALYSIS

11.1 INTRODUCTION 11.1-1
11.2 TRANSIENT LOADS 11.2-1
11.3 NONLINEAR ELEMENTS 11.3-1
11.4 INTEGRATION OF COUPLED EQUATIONS 11.4-1
11.4.1 Solution Technique and Procedure 11.4-1
11.4.2 The CONTINUE Feature for Coupled Equations 11.4-14
11.5 INTEGRATION OF UNCOUPLED LINEAR EQUATIONS 11.5-1
11.6 TRANSIENT ANALYSIS BY THE FOURIER METHOD 11.6-1
11.6.1 Transformation of Loads 11.6-2
11.6.2 Calculation of Frequency Response 11.6-4
11.6.3 Inverse Transformation of the Response 11.6-4
11.6.4 Practical Considerations for Use of the Fourier Methods 11.6-6

12. FREQUENCY RESPONSE AND RANDOM ANALYSIS

12.1 INTRODUCTION 12.1-1
12.2 FREQUENCY RESPONSE 12.2-1
12.3 RANDOM ANALYSIS 12.3-1

13. COMPUTER GRAPHICS

13.1 STRUCTURE PLOTTER 13.1-1
13.1.1 Structure Plotter Coordinate System and Orthographic Projection 13.1-1

TABLE OF CONTENTS (Continued)

Section Page No.

13.1.2 Perspective Projection 13.1-3
13.1.3 Stereoscopic Projection 13.1-4
13.1.4 Projection Plane-to-Plotter Transformations 13.1-4
13.1.5 Deformation Scaling 13.1-5
13.1.6 Structure Plotter Examples 13.1-5
13.2 CURVE PLOTTER 13.2-1
13.3 MATRIX TOPOLOGY PLOTTER 13.3-1

14. SPECIAL MODELING TECHNIQUES

14.1 REPRESENTATION OF PART OF A STRUCTURE BY ITS VIBRATION MODES 14.1-1
14.2 REPRESENTATION OF CONTROL SYSTEMS 14.2-1

15. ERROR ANALYSIS

15.1 SELECTION OF SIGNIFICANT FIGURES BASED ON ROUND-OFF ERRORS 15.1-1
15.2 MODELING ERRORS IN THE BENDING OF PLATE STRUCTURES 15.2-1
15.2.1 Triangular Element 15.2-1
15.2.2 Quadrilateral Element 15.2-1
15.2.3 Higher Order Triangular Plate Bending Element 15.2-2
15.3 MODELING ERRORS IN MEMBRANE ELEMENTS 15.3-1
15.3.1 Quadrilateral Membrane Elements 15.3-1
15.3.2 Linear Strain Triangular Membrane Element 15.3-1
15.4 DECOMPOSITION ERROR CHECKS 15.4-1

16. INTERACTION BETWEEN STRUCTURES AND FLUIDS

16.1 COMPRESSIBLE FLUIDS IN AXISYMMETRIC TANKS 16.1-1
16.1.1 Approach 16.1-1
16.1.2 Fluid Field Equations 16.1-3
16.1.3 Boundary Conditions 16.1-7
16.1.4 Fluid Elements 16.1-14
16.1.5 Boundary Matrices 16.1-27
16.1.6 Applications 16.1-35
16.2 COMPRESSIBLE FLUIDS IN ROTATIONALLY SYMMETRICAL CAVITIES WITH SLOTS 16.2-1
16.2.1 Formulation of the Problem 16.2-1

TABLE OF CONTENTS (Continued)

Section Page No.

16.2.2 Fluid Elements for the Slots 16.2-2

16.2.3 Corrections for the Effects of Finite Slot Width 16.2-9

16.2.4 Recovery of Velocity Components 16.2-13

17. AEROELASTIC ANALYSIS

17.1 INTRODUCTION 17.1-1

17.2 DATA INPUT AND GENERATION 17.2-1

17.3 INTERCONNECTION OF STRUCTURE WITH AERODYNAMICS 17.3-1

17.3.1 Theory for Surface Splines 17.3-1

17.3.2 Theory of Linear Splines 17.3-3

17.3.3 Attachment of Splines with Elastic Springs 17.3-6

17.3.4 Rigid Arms on Linear Splines 17.3-7

17.3.5 Coordinate Systems and Constraints 17.3-7

17.4 FLUTTER SOLUTION TECHNIQUES 17.4-1

17.4.1 The K-method of Flutter Solution 17.4-1

17.4.2 The KE-method of Flutter Solution 17.4-2

17.4.3 The PK-method of Flutter Solution 17.4-3

17.5 UNSTEADY AERODYNAMIC THEORIES 17.5-1

17.5.1 Doublet-Lattice Method 17.5-1

17.5.2 Subsonic Wing-Body Interference Theory 17.5-2

17.5.3 Mach Box Method 17.5-6

17.5.4 Strip Theory 17.5-8

17.5.5 Piston Theory 17.5-9

17.6 AEROELASTIC RIGID FORMATS 17.6-1

14

1. PROGRAM ORGANIZATION

1.1 OVERVIEW OF THE PROGRAM

NASTRAN is a finite element computer program for structural analysis that is intended for general use. As such it must answer to a wide spectrum of requirements. The program must be efficient, versatile and convenient to use. It must be standardized to permit interchange of input and output between different users. It must be structured to permit future modification and extension to new problem areas and to new computer configurations without major redevelopment.

The intended range of applications of the program extends to almost every kind of structure and to almost every type of construction. Structural elements are provided for the specific representation of the more common types of construction including rods, beams, shear panels, plates, and shells of revolution. More general types of construction are treated by combinations of these elements and by the use of "general" elements. Control systems, aerodynamic transfer functions, and other nonstructural features can be incorporated into the structural problem.

The range of analysis types in the program includes: static response to concentrated and distributed loads, to thermal expansion and to enforced deformation; dynamic response to transient loads, to steady-state sinusoidal loads and to random excitation; determination of real and complex eigenvalues for use in vibration analysis, dynamic stability analysis, and elastic stability analysis. The program includes a limited capability for the solution of nonlinear problems, including piecewise linear analysis of nonlinear static response and transient analysis of nonlinear dynamic response.

NASTRAN has been specifically designed to treat large problems with many degrees of freedom. The only limitations on problem size are those imposed by practical considerations of running time and by the ultimate capacity of auxiliary storage devices. The program is decidedly not a core program. Computational procedures have been selected to provide the maximum obtainable efficiency for large problems.

Research was conducted during the design of the program in order to ensure that the best available methods were used. The areas of computer program design that are most sensitive to state-of-the-art considerations are program organization and numerical analysis. The organizational demands on the program design are severe in view of the multiplicity of problem types and user conveniences, the multiplicity of operating computer configurations, the requirement for large problem capability, the requirement for future modification, and the requirement for

PAGE 15 INTENTIONALLY BLANK

16

PRECEDING PAGE BLANK NOT FILMED

responsiveness to improvements in programming systems and computer hardware. The organizational problems have been solved by applying techniques that are standard in the design of computer operating systems but have not, as yet, been extensively used in the design of scientific applications programs. The main instrument of program organization in the program is an executive system that schedules the operating sequence of functional modules and that plans and allocates the storage of files. An important aspect of the executive routine concept used in NASTRAN is that it greatly reduces the cost of program coding and checkout by eliminating most module interface problems and by reducing the remainder to a form that permits systematic treatment.

Most difficulties in numerical analysis arise in connection with three basic implicit operations: matrix decomposition (or inversion), eigenvalue extraction, and integration of differential equations. The major difficulties that occur in the application of these operations to large problems are excessive computing time, error accumulation and instability. Many methods that work well with small or moderate sized problems are not acceptable for large problems.

The method employed for matrix decomposition is especially important due to its extensive use as a base for the other two implicit operations. The method that is employed in the program takes maximum advantage of matrix sparsity and bandedness. The latter aspect is particularly important due to the enormous gain in efficiency that accrues when banding techniques are properly employed by the user in setting up problems for the displacement method.

In general the solution time for a large structural analysis of any type can be greatly reduced by taking full advantage of the sparsity and bandwidth of the matrices that describe the structural problem. Other means, in addition to the matrix decomposition routine mentioned above, have been used to improve efficiency for large problems. These include storing sparse matrices in packed form, the avoidance of operations that reduce sparsity or destroy bandwidth, well designed Input/Output strategies, the use of advanced techniques for eigenvalue extraction, and specially tailored numerical integration algorithms.

The needs of the structural analyst have been considered in all aspects of the design of the program. The first thing to be remembered is that, in view of the wide range of possible applications of the program, we do not know exactly what these needs may be. For this reason a high degree of flexibility and generality has been incorporated into certain areas of the program. For example, in addition to the usual list of structural elements that refer to specific types of construction, the user is provided with more general elements that may be used to construct any type of special element, to represent part of a structure by deflection influence coefficients, or to represent part of a structure by its vibration modes. For the more conventional types of structural analysis, the user is presented with a large number of convenience features, including plotting routines, which are definite necessities for large problems.

A major difficulty that the user faces in the solution of large problems is the avoidance of errors in the preparation of input data. Card formats and card ordering are made as simple and flexible as possible in NASTRAN in order to avoid errors caused by trivial violations of format rules. A number of aids for the detection of legal but incorrect data are also provided.

The problems that can be solved by NASTRAN include the following general classes:

1. Static Structural Problems
2. Elastic Stability Problems
3. Dynamic Structural Problems
4. General Matrix Problems
5. Heat Transfer Problems
6. Aeroelasticity Problems

Each general problem class is further subdivided into case types which differ with regard to the type of information desired, the environmental factors considered, or the method of analysis. The mathematical computations required to solve problems are performed by subprogram units called functional modules. Each case type requires a distinct sequence of functional module calls that are scheduled by the Executive System.

For structural problem types the sequence of module calls and hence the general method of solution is established internally for each case type according to a rigid format stored in the Executive System. Execution of a structural problem proceeds in one run to final solution, or, at the option of the user, to a desired intermediate point.

A more flexible procedure is provided for the solution of general matrix problems. All of the matrix operations (such as addition, multiplication, triangular decomposition, and eigenvalue extraction) used in the program can be directly addressed by the user according to a system of macro instructions called DMAP(for Direct Matrix Abstraction Program). The user constructs a chain of DMAP instructions in order to effect the solution of general matrix problems.

1.2 THE NASTRAN EXECUTIVE SYSTEM

1.2.1 Introduction

The overall effectiveness of a general purpose program depends in large measure on how well the available programming techniques have been employed in the design of its organizational and control features. It may, therefore, be useful to precede the usual treatment of the engineering and mathematical aspects of the program with a discussion of a relatively unfamiliar feature of general purpose programs, namely the Executive System.

NASTRAN has been designed according to two classes of criteria. The first class relates to functional requirements for the solution of an extremely wide range of large and complex problems in structural analysis with high accuracy and computational efficiency, which are met by developing advanced mathematical models of the physical phenomena and incorporating their computation algorithms into the program. The second class of criteria relates to the operational and organizational aspects of the program. These aspects are somewhat divorced from structural analysis itself; yet they are of equal importance in determining the usefulness and quality of the program. Chief among these criteria are:

1. Simplicity of problem input deck preparation.
2. Minimization of chances for human error in problem preparation.
3. Minimization of need for manual intervention during program execution.
4. Capability for step by step problem solution, without penalty of repeated problem set up.
5. Capability for problem restart following unplanned interruptions or problem preparation error.
6. Minimization of system overhead, in the three vital areas:
   a. Diversion of core storage from functional use in problem solution.
   b. Diversion of auxiliary storage units from functional to system usage.
   c. System housekeeping time for performing executive functions that do not directly further problem solution.
7. Ease of program modification and extension to new functional capability.

PRECEDING PAGE BLANK NOT FILMED

PAGE 19 INTENTIONALLY BLANK

8. Ease of program extension to new computer configurations and operating systems, and generality in ability to operate efficiently under a wide set of configuration capabilities.

The second class of eight objectives is achieved in NASTRAN through modular separation of functional capabilities, organized under an efficient, problem-independent executive system. This approach is absolutely essential for any complex multioperation, multifile application program such as NASTRAN. To see this, one must examine the implications of modularity in program organization.

Any application computer program provides a selection of computational sequences that are controlled by the user through externally provided options and parameter values. Since no user will wish to observe the result of each calculation, these options also provide for the selection of the data to be output. In addition to externally set options, internal decision switches whose settings depend upon tests performed during the calculations will control the computation sequences. There is, therefore, a natural separation of computations into functional blocks. The principal blocks are called functional modules; modules themselves, of course, may and usually must be further organized on a submodular basis.

Despite this separation, however, it is clear that modules cannot be completely independent, since they are all directed toward solution of the same general problem. In particular, they must intercommunicate data between themselves. The principal problem in organizing any application program, large or small, is designing the data interfaces between modules.

For small programs, the standard techniques are to communicate data via subroutine calling sequences and common data regions in core storage. For programs that handle larger amounts of data, auxiliary storage is used; however, strict specifications of the devices used and of the data record formats are usually imposed. The penalty paid is that of "side effects". A change in a minor subroutine initiates a modification of the data interfaces that propagates through the entire program. When the program is small, these effects may not be serious. For a complex program like NASTRAN, however, they may be disastrous.

This problem has been solved in NASTRAN by a separation of system functions, performed by an executive routine, from problem solution functions, accomplished by modules separated strictly along functional lines. Each module is independent from all other modules in the sense that modification of a module, or addition of a new module, will not, in general, require modification of other modules. Even so, programming constraints on module development are minor. The essential restrictions are:

1. Modules may interface with other modules only through auxiliary storage files that contain data blocks.
2. Since the availability of the auxiliary files required for the execution of a module depends on the execution of other modules, no module can specify or allocate files for its input or output data. All auxiliary storage allocation is reserved as an executive function.
3. Modules operate as independent subprograms, and may not call, or be called by, other modules. They may be entered only from the executive routine.
4. Modules may interface with the executive routine through a parameter table that is maintained by the executive routine. User specified options and parameters are communicated to modules in this way. The major line of communication is one-way, from user to executive routine to module. However, in addition, an appreciable two-way communication from module back to executive routine (and, therefore, to other modules) is permitted via the parameter table.

No other constraints, except those imposed by the resident compilers and operating systems, are required for functional modules.

The essential functions of the executive system are:

1. To establish and control the sequence of module executions according to options specified by the user.
2. To establish and communicate values of parameters for each module.
3. To allocate files for all data blocks generated during program execution and perform input/output to auxiliary files for each module.
4. To maintain a full restart capability for restoring a program execution after either a scheduled or unscheduled interruption.

Each of these functions is essentially independent of any particular feature of structural analysis and applies to the operational control of any complex multimodule, multifile application program. The executive system is open-ended in the sense that it can accommodate an essentially unlimited number of functional modules, files, and parameters. Modification of the executive system necessary for modification or extension of functional modules is restricted to changes in entries in control tables stored within the executive routine.

A description of the way in which these objectives and functional capabilities are accomplished by the NASTRAN Executive System is included in the following sections.

1.2.2 Executive Operations During the Preface

Program execution is divided into a preface and the program body proper. During the preface the NASTRAN Executive System analyzes and processes the data which define user options regarding the structural problem to be solved and organizes the overall problem solution sequence. The sequence of operations during the preface is presented in Figure 1 and is described in detail in succeeding subsections. During the program body proper, the NASTRAN Executive System controls the step-by-step problem solution sequence.

1.2.2.1 Generation of the Initial File Allocation Tables

Two file allocation tables are maintained by the NASTRAN Executive System. One table defines the files to which data blocks generated during solution of the problem will be allocated. The second table includes files to which permanent executive data blocks, such as the New Problem Tape, the Old Problem Tape, the Plot Tape, and the User's Master File are assigned.

The New Problem Tape will contain those data blocks generated during the solution that are necessary for restarting the problem at any point. The Old Problem Tape contains the data blocks saved from some previous execution that may serve to bypass steps in the solution of the new problem. The Plot Tape includes output data and plotting instructions in a form that will be accepted by an automatic plotter selected by the user. The User's Master File is a permanent collection of useful information, such as material properties, that may be used to generate input data.

The generation of the file allocation tables is an operation that depends on the particular computer model being used since direct interface with the operating system of the computer must be made. The routine which accomplishes this function interrogates file tables that are located in the nucleus of the computer's own resident operating system. Files which are available for use by the NASTRAN program are reserved and the unit numbers are stored in the NASTRAN file allocation tables. An indication of which units are physical tapes is noted. If the number of files available is insufficient, an error message is generated and the run is aborted.

1.2.2.2 Analysis of the Executive Control Deck

The first purpose of executive control is to provide a level of regulation for the many options within NASTRAN. At this level the executive distinguishes between the broad approaches to problem solution, e.g., between a matrix abstraction approach by the analyst or a rigid format approach according to problem class. Also at this level, the executive distinguishes between several operational modes, e.g., a first attempt, a continuation, or a modification. Certain other functions of a general nature are convenient to include with the executive control such as problem identification, selection of a level of diagnostics, and the estimation of solution time.

The executive control deck includes cards which describe the nature and type of the solution to be performed. These include an identification of the problem, an estimated time for solution of the problem, a selection of an approach to the solution of the problem, a restart deck from a previous run if the solution is to be restarted, an indication of any special diagnostic printout to be made, and a specification of whether execution of the problem is to be completed in a single run, or whether execution will be stopped (check-pointed) at some intermediate step.

Each of the cards comprising the executive control deck is read and analyzed. Depending on the card, information is either stored in various executive tables maintained in core storage or written in a Control Table on the New Problem Tape for further processing during a later phase of the preface.

1.2.2.3 Processing of the Case Control Deck

When the rigid format solution route is selected, further details of control are provided by the 'Case Control' portion of the executive. In effect, the analyst can manipulate his problem by means of entries he inserts in the Case Control. He can make choices amongst the sets of data representing different physical situations which are allowed to be assembled in the Bulk Data portion of the problem input. Here also the analyst can regulate his output. Fundamental to the method of control in this section is the notion of sets. Boundary conditions, loading cases, and output selections are controlled by set selection.

The case control deck includes cards that indicate the following options: selection of specific sets of data from the bulk data deck (i.e., from the data deck that describes the details of a problem), selection of printed or punched output, definition of subcases, and the definition of plots to be made.

The case control deck is read and processed. Information defining data set selection, output format selection and subcase definition is written in the Case Control data block. Information defining plot requests is written in the Plot Control data block.

If the problem is a restart, a comparison with the Case Control data block from the previous run (stored on the Old Problem Tape) is made. Differences are noted in an executive restart table.

1.2.2.4 Sorting of the Bulk Data

In NASTRAN the input to the mathematical operations performed in functional modules is provided in the form of previously organized data blocks. The data blocks derive from two sources: those that derive from the bulk input data and those that are generated as output from previous functional modules. Those that derive from the bulk data are organized into data blocks by the IFP routine, but prior to the execution of IFP, XSØRT sorts the bulk data. Operation of the XSØRT routine is influenced by the type of run. If the run is a cold start (that is, an initial submittal for a given job) the bulk data is read from the system input unit or the User's Master File, is sorted, and is written on magnetic files in preparation for problem execution. If the analyst wants to provide for a future restart, the SORT routine prepares a file on the New Problem Tape which contains the sorted bulk data. If the run is a restart, the bulk data is copied from the Old Problem Tape with the addition of any changes from the system input unit.

An echo of the unsorted bulk data is given if requested. Similarly, the sorted bulk data is echoed on request.

Since the collating sequence of alphanumeric characters varies from computer to computer, the sort routine converts all characters to an internal code prior to sorting. Following the sort, the characters are reconverted. In this way, the collating sequence is made computer independent.

The algorithm used by the sort routine is biased toward the case where the data is in sort or nearly in sort. Consequently, bulk data decks which are nearly in sort will be processed efficiently by the routine.

The sorted bulk data is read from the New Problem Tape by the Input File Processor. Each of the cards is checked for correctness of format. If any data errors are detected, a message is written and a switch is set to terminate the run at the conclusion of the preface.

Processing of the bulk data cards depends on the type of information on the card. Each set of data cards of the same type is written as one logical record in the data block to which the card has been assigned.

1.2.2.5 General Problem Initialization

The general problem initialization is the heart of the preface. Its principal function is to generate the Operation Sequence Control Array (ØSCAR) which defines the sequence of operations for an entire problem solution. The ØSCAR consists of a sequence of entries, with each entry containing all of the information required to execute one step of the problem solution. The ØSCAR is generated from information supplied by the user in the executive control deck.

If the problem is a restart, the restart dictionary (contained in the Control Table) and the executive restart table are analyzed to determine which data blocks are needed to restart the solution and which operations need to be executed to complete the solution.

To aid in efficient assignment of data blocks to files, two ordinals are computed and included with each data block in each entry of the ØSCAR. These ordinals are the ØSCAR sequence number indicating when the data block is next used and the ØSCAR sequence number indicating when the data block will be used for the last time.

When generation of the ØSCAR is complete, it is written on the PØØL (an executive data block). If the problem is a restart, data blocks needed for the current solution are copied from the Old Problem Tape to the PØØL, augmented by entries to provide for new current requirements.

1.2.3 Executive Operations During Problem Solution

1.2.3.1 Sequence Monitor

When the preface has been completed, solution of the problem is initiated. The solution is controlled by the sequence monitor.

The sequence monitor reads an entry from the ØSCAR which defines one step in the problem solution in terms of the operation to be performed, data blocks required for input, data blocks to be output, scratch (i.e., temporary) files required, and parameters. A status table is generated which relates the names of data blocks required for operation to the position in the file allocation table where information about the data block is contained. When the status table is complete and the parameters required for the operation have been retrieved from the parameter storage table, the appropriate functional module is called to execute the operation.

1.2.3.2 Segment File Allocation

The segment file allocator is the administrative manager of data blocks for NASTRAN. All large modern computers have sufficient auxiliary storage to accommodate the needs of NASTRAN. The number of separate files into which the storage can be divided is, however, severely limited on most computers. In general, the number of data blocks required for solution of a problem far exceeds the number of files available, so that the assignment of data blocks to files is a critical operation for efficient execution of NASTRAN.

The segment file allocator is called whenever a data block is required for execution of an operation but is not currently assigned to a file. When the segment file allocator is called, it attempts to allocate files for as much of the problem solution as possible. This depends on the type of problem, the number of files available, and the range of use of the data blocks.

The segment file allocator reads entries from the ØSCAR from the point of current operation to the end of the problem solution. A table is assembled in which information about data blocks, including their next use and their last use, is stored. Data blocks which are currently assigned to files but are no longer required for problem solution are deleted. In certain cases, when the range of use of a data block is large, it may not be possible to allocate a file to the data block throughout its entire range of use. In this case, pooling of the data block into a single file with other data blocks is required so that the file to which the data block was assigned may be freed for another allocation. In general, those data blocks whose next use is furthest from the current point are pooled.

When the segment file allocator has completed its task, a new file allocation table has been generated. This table is used until the solution again reaches a point where a data block is required to execute an operation but is not assigned to a file.

1.2.3.3 Input/Output Operations

All input/output operations in NASTRAN (except reading data from the system input file or writing data on the system output file) are controlled by a collection of executive routines called GIN0 (General Input Output) which act as a buffer between the NASTRAN functional modules and the operating system of the computer. This design feature eliminates computer dependent code from the functional module programs which are, consequently, written exclusively in FORTRAN. The use of computer dependent code for the selection of the operating system routines to accomplish the actual input/output functions is isolated to a single routine within GIN0.

1.2.3.4 Other Executive Operations

Additional operations in support of a problem solution which are performed by the NASTRAN Executive System include checkpoint, purge, equivalence and save.

The checkpoint routine copies data blocks required for problem restart onto the New Problem Tape and makes appropriate entries in the restart dictionary.

The purge and equivalence routines change the status of data block entries in the file allocation table. They are called whenever the nature of a given problem requires less than the full generality provided within NASTRAN, thereby permitting some computational steps to be bypassed.

The save routine stores the values of parameters in the parameter storage table where they are retrieved for subsequent use by the sequence monitor.

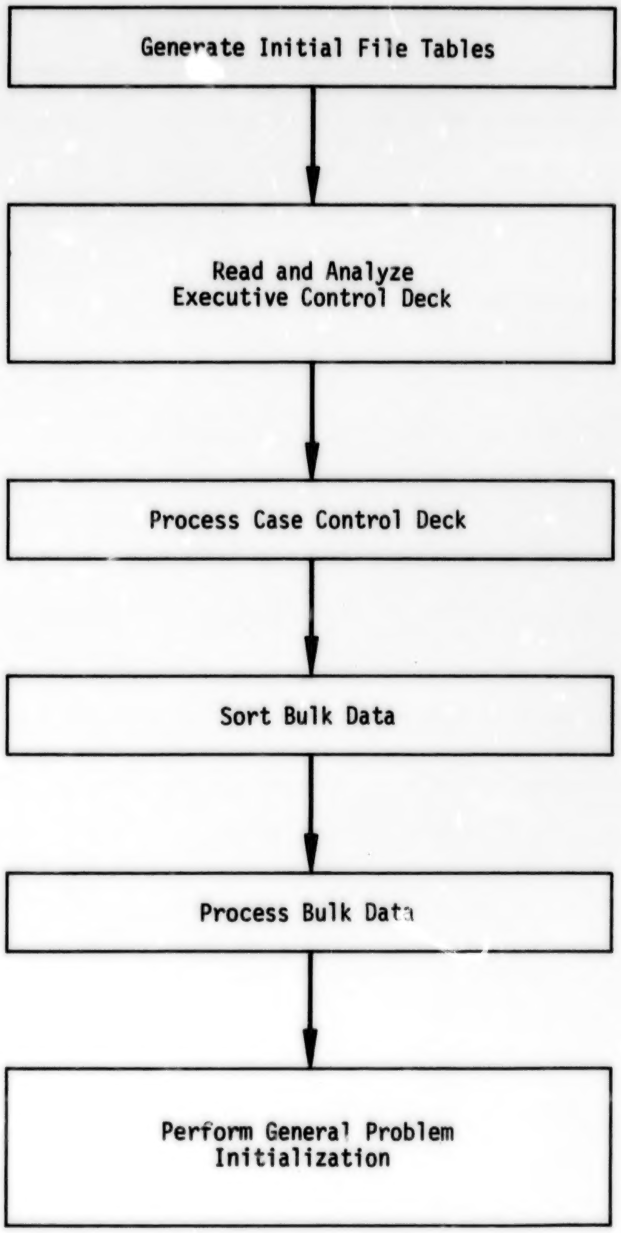

Generate Initial File Tables
Read and Analyze Executive Control Deck
Process Case Control Deck
Sort Bulk Data
Process Bulk Data
Perform General Problem Initialization

Figure 1. Flow of operations during the preface.

1.3 USER CONTROL OF PROBLEM EXECUTION

All general purpose programs have formal procedures by which the user controls the calculations that are performed. In NASTRAN several modes of operation and a large number of options within each mode are provided to the user. A short discussion of these matters is presented here for completeness. More extensive treatment will be found in the User's Manual.

During the solution of a problem, the NASTRAN executive system calls a sequence of functional modules that perform the actual calculations, as explained in the preceding section. Two general types of solution are provided: solution by Rigid Format according to a sequence of module calls built into the program; and solution according to a sequence of module calls generated by the user. The latter capability is provided in order to make the program's matrix routines available for general use and also to provide the sophisticated user with the means for solving structural problems with features not accounted for in any of the built-in module sequences. It is intended, however, that the great majority of structural problems will be solved via the rigid formats.

There are, at present, a total of twelve rigid formats in NASTRAN with provision for adding an unlimited number in the future. Each corresponds to a particular type of solution or to a particular method of analysis, such as: Static Analysis, Buckling Analysis, Direct Transient Response, Modal Transient Response, etc. The five Rigid Formats associated with static analysis are described in Section 3.2. The seven Rigid Formats associated with dynamic analysis are described in Section 9.1.

Each rigid format consists of two parts. The first is a sequence of instructions (including instructions for Executive operations as well as for Functional Module operations) that is stored in tables maintained by the Executive System. The second part is a set of restart tables that automatically modify the sequence of instructions to account for any changes in the input data when a restart is made after partial or complete execution of a problem. The restart tables can accommodate a change of rigid format such as occurs, for example, when vibration modes are requested for a structure that was previously analyzed statically. The restart tables are, as can be imagined, quite extensive and their generation constitutes a significant part of the effort expended in developing a rigid format. They are, however, one of the more important cost-saving features of NASTRAN.

Many options are available with each rigid format. One such option removes the possibility of branching back to previously executed functional modules, and it should, of course, be exercised only when it is known in advance that looping will not occur. This option permits the Executive System to discard files that would otherwise be saved. Other options define the subcases to be executed and the desired output formats, see Section 1.2.2.3.

It is also possible for the user to modify a rigid format via the ALTER feature described in Section 2 of the User's Manual. Typical uses of the ALTER feature are to schedule an exit at an intermediate point in a solution for the purpose of checking intermediate output, to schedule the printing of a table or a matrix for diagnostic purposes, and to add or delete a functional module from the sequence of operating instructions.

For more extensive modifications the user can write his own sequence of executive instructions. The system by which this is done is called DMAP (for Direct Matrix Abstraction Program). DMAP is a user-oriented programming language of macro instructions which, like FORTRAN, has many rules which must be followed to be interpretable by NASTRAN. DMAP is also used in the construction of rigid formats, which differ from user-generated sequences mainly in that restart tables are provided.

The rules for generating a DMAP sequence are explained in Section 5 of the User's Manual. The DMAP sequence itself consists of a series of statements consisting of Executive Operation instructions and Functional Module calls. Each statement contains the name of the instruction (or Functional Module), the names of the input data blocks, the names of the output data blocks, and the names and values of parameters. Typical examples of parameter usage are to indicate whether an operation is to be performed with single or double precision arithmetic, which mathematical method will be used (when there are options), or the desired format of the output.

The names of some of the executive operations are BEGIN; CHKPNT (used when it is desired to copy data blocks onto the Problem Tape in case an unscheduled restart is necessary); FILE (used to save an intermediate data block); REPT (used to provide looping capability); PURGE (used to prevent storage of data blocks); and END.

The functional modules belong to one of the following categories: structural modules; matrix operations; utility modules; and user modules. The Structural Modules are the main subprograms of NASTRAN. Some examples of structural modules, taken from dynamic analysis, are: READ (Real eigenvalue analysis); GKAM (Modal dynamic matrix assembler); TDR (Transient Dynamic Response); and DDR (Dynamic Data Recovery). The Matrix Operations (add, multiply, transpose, etc.) that are available to the user of NASTRAN are described in Section 2. The Utility Modules are mainly concerned with the formats of output data. The User Modules are dummy modules that provide the user with the ability to write new functional capability that will automatically be recognized by the executive system.

The usual methods of output for NASTRAN are the operating system print or punch files and the NASTRAN plot tapes. Procedures for normal output selection are described in Section 2.3 of the User's Manual. The printing of tables or matrices generated by NASTRAN is controlled by a group of Utility Modules described in Section 5.3.2 of the User's Manual. In many cases, it is desirable to save matrices and tables for use in restart operations. When using rigid formats, it is possible to save preselected tables and matrices by using the Checkpoint option described in Section 2.2 of the User's Manual. Checkpointed files are written on the New Problem Tape. It is also possible for the user to save selected matrices on tape by inserting one of the User Modules described in Section 5.3.3 of the User's Manual into the DMAP sequence by means of the ALTER option.

The usual method of input for NASTRAN is the operating system card reader. When performing restarts, the New Problem Tape from a previous run is redesignated as the Old Problem Tape and used as an additional source of input. Tapes that have been prepared with User Modules on previous runs can also be used as additional input sources by inserting one of the input User Modules into the DMAP sequence by means of the ALTER option.

2. MATRIX OPERATIONS

2.1 ELEMENTARY OPERATIONS

2.1.1 Introduction

The operations to be considered (matrix add, multiply, transpose, partition and merge) are sufficiently elementary that the formal mathematical procedures which accomplish them may safely be assumed to be well-known to all readers of the Theoretical Manual. What is not likely to be known is the corresponding sequence of physical data manipulations that are performed by the computer. Such matters are not usually considered to be required reading for users or for others with an interest in "the theory"; they are, accordingly, buried in the programmers' manual as reference material for maintenance and modification of the program. This practice is not followed here because the success or failure of NASTRAN depends, to a far greater extent than for smaller programs, on the efficiency of the subroutines that perform the basic matrix operations. All matrix operations in NASTRAN are performed by specially designed subroutines.

Questions regarding accuracy, which is an equally important aspect of numerical calculation, fall into two categories: those that relate to analytical approximations, such as occur in iterative solutions, and those that relate to simple round-off error accumulation. Elementary matrix operations do not involve analytical approximations. Nor do the triangular decomposition of matrices and the solution of simultaneous linear equations, described in Sections 2.2 and 2.3. The errors that occur in eigenvalue extraction and in numerical integration due to analytical approximation are discussed in the sections dealing with those topics.

Trigonometric and other elementary irrational functions are evaluated by library subroutines provided by the manufacturer of the computer, who guarantees them to be accurate.

The effects of round-off error accumulation in structural analysis are treated in Section 15.1, where reasons are presented for adopting double precision arithmetic (54 or more bits) in critical calculations. No other measures are employed in NASTRAN for combating round-off error accumulation. The usual measures of this sort (e.g., rounding rather than truncating arithmetic results, or accumulating sums by starting with the smallest numbers) are only mildly effective and have the disadvantages that they require machine language coding, or that they substantially increase running time, or both.

From the viewpoint of data processing, the computer has two main parts: a central processor that contains an arithmetic unit and a randomly accessible memory device (core storage) with very

PAGE 33 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

short access time; and a collection of peripheral storage devices (tapes, disks and/or drums) with high capacity but relatively long access times. In general the data contained on the peripheral storage devices can be accessed effectively by the central processor only in relatively large blocks, due to the time required to locate the first word in any record. Thus, from the viewpoint of matrix algebra, data should be sequentially read from and written on peripheral storage devices as one or two-dimensional arrays. An important convention employed in NASTRAN is that all matrices are stored on peripheral devices by columns. This fact is important to the discussion of the matrix multiply and transposition subroutines described below.

It is assumed, in the design of NASTRAN, that a typical matrix is so large that it cannot all be held in (high-speed) core storage at any one time, even if it is a sparse matrix that is expressed in packed form (i.e., by means of its nonzero elements and their row-column indices). In such situations, the computing time tends to be dominated by the relatively slow rate of data transfer from peripheral storage to core storage, and optimum computing strategies are designed to minimize the number of data transfers.

The time to transfer a sparse matrix from peripheral storage to the central processor will be decreased if only the nonzero terms are stored. The matrices in NASTRAN are packed in nonzero strings in the following manner. The record for each column begins with a three-word header. This is followed by an integer (fixed-point number) describing the position (row index) of the first nonzero term and by a second integer describing the number of consecutive nonzero terms in the string. The integers are followed in consecutive locations by the floating point numbers describing the values of the nonzero terms in the string. The remaining nonzero strings follow in order until the end of the column is reached. The data record describing a typical column will appear as follows:

I, I, I, 2, 2, (X,X), 8, 3, (X,X,X), 17, 1, (X), 27, 1, (X), E.

The three I's are the header for the column. The X's are the numerical values of terms, and E indicates the end of the record. The nonzero terms in the column are the 2nd, 3rd, 8th, 9th, 10th, 17th, and 27th. Once the record is transferred to core storage, it may, if required, be fully expanded by addition of the zero terms.

In the case of triangular factors, the integers describing the row position and the number of consecutive nonzero terms, are placed at the end, as well as the beginning of each nonzero string.

This manner of storing sparse matrices allows the matrix to be read backward in the same manner as it is read forward, and thereby allows for improved efficiency in the backward substitution part of equation solution operations.

2.1.2 Matrix Multiplication

The multiplication of large matrices can be a time consuming operation. If the matrices are full, then the time to multiply two matrices of order nxm and mxr is proportional to nmr. If the matrices are sparse, but no attempt is made to take advantage of the sparsity, the running time will be the same as if the matrices were full.

Most of the matrices used in structural analysis are initially very sparse. They may, however, become relatively dense as the result of transformations. Consequently, the NASTRAN program requires a matrix multiplication routine that works well for sparse matrices as well as for full matrices.

The matrix multiplication routine in NASTRAN provides two alternative methods of matrix multiplication. Both of the methods take advantage of sparsity in different ways. The second method might be described as a truly sparse matrix method in that only the nonzero terms in either the left-hand or the right-hand matrices are processed. The method which results in the minimum execution time is automatically selected by the routine.

For the discussion which follows, the general multiply-add form, [D] = [A][B]+[C], is assumed.

In Method One, core storage is allocated to hold as many columns of [B] and [D] in unpacked form as possible (columns of [C] being read initially into the storage space for [D]). The [A] matrix is read interpretively one nonzero element at a time. For each nonzero element in [A], all combinatorial terms for columns of [B] currently in core are computed and accumulated in the storage for [D]. Let $a_{i\ell}$ be a nonzero element of [A] and $b_{\ell j}$ be an element of [B]. The formula for an element of [D] is

$$d_{ij} = \sum_{\ell} a_{i\ell} b_{\ell j} + c_{ij} \quad , \tag{1}$$

where j runs across the columns of [B] and [D] currently in core. At the completion of one complete pass of the [A] matrix through the central processor, the product is completed to the extent of the columns of [B] currently in core. The process is repeated until the [B] matrix is exhausted. It may be seen that the number of passes of the [A] matrix equals the total number of columns of [B] divided by the number of columns of [B] that can be held in core at one time. Method One is effective if the number of columns of [B] is not large, e.g., when [B] is a small number of load vectors. Method One is also more effective than Method Two when [B] is a dense matrix.

PAGE 2.1-2 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

In Method Two, only a single term of the [B] matrix is required in main memory at any one time. One full column in unpacked form of the partially formed [D] matrix is also stored in core at the same time. The remaining storage is allocated to as many columns of [A] in packed form as can be stored, i.e., only nonzero terms and corresponding row positions are stored. For the columns of [A] in storage at one time, the [B] matrix is passed through the central processor, column by column, forming partial answers on each pass.

Each column of [B] forms partial answers which are added to the corresponding column of [D]. As may be seen from Equation 1, only the elements in the rows of $b_{\ell j}$ corresponding to the columns of $a_{i\ell}$ currently held in core are used. After all columns of [B] have been processed once, new columns of [A] are placed in core and the [B] matrix is passed through again. The process is repeated until all the columns of [A] have been used.

In Method Two the [A] matrix is passed through core once and the number of passes of the [B] matrix equals the total number of columns of [A] divided by the number of columns of [A] that can be held in core in packed form at one time. The number of passes of the [B] matrix is the controlling factor in determining computing time. If the [A] matrix is large and sparse, the number of passes of the [B] matrix in Method Two will typically be less than five. In Method One, on the other hand, the number of passes of the [A] matrix will be much larger if the number of columns of [B] is large. The reason is that, in Method One, the columns of the [B] matrix are not stored in packed form, whereas, in Method Two, the columns of the [A] matrix are stored in packed form.

Both methods one and two include variations for premultiplication of a matrix by the transpose of another matrix, $[D] = [A]^T[B]+[C]$, where [A] is stored by columns. This is done in order to avoid transposing the [A] matrix, which is by no means trivial (see Section 2.1.4). In fact, the second matrix multiply method provides an efficient means for matrix transposition of sparse matrices, by setting [B] = [I] and [C] = 0.

A third option is provided for the transpose case in order to efficiently handle the case of [B] sparse and [A] dense. The operations for method three are similar to those described for the nontranspose case of method two, except the columns of A (rows of $[A]^T$) are held in unpacked rather than packed form. In the transpose case for method two the computing time is proportional to the density of the [A] matrix, whereas in method three the computing time is proportional to the density of the [B] matrix. A nontranspose option is not needed for method three as the computing time for the nontranspose option in method two is proportional to the product of the densities of the [A] and [B] matrices.

2.1.3 Matrix Addition

The addition routine computes the general matrix sum,

$$[C] = a[A] + b[B] \quad , \tag{2}$$

where a and b are scalars and [A] and [B] matrices. Special provision is made for the case b = 0, to allow scalar multiplication. No compatibility of types (such as single or double precision, real or complex numbers) between a, b, [A], and [B] is required. The nonzero terms of [A] and [B] are read interpretively one nonzero element at a time. The appropriate sum is formed into [C] and immediately transferred to peripheral storage. The required amount of core storage is very small.

2.1.4 Matrix Transposition

The transposition of large matrices is a distressingly awkward operation. The optimum strategy depends on the location of the nonzero terms, the density of the matrix, and its size. The NASTRAN algorithm which is used in the transposition of dense matrices is described below. Sparse matrices are transposed by the matrix multiply subroutine (see above).

If the matrix order is i x j and if only a fraction of the matrix may be held in core at one time, the usual technique is to read the whole matrix from a peripheral storage device, saving, in core, the elements from the first R rows of the matrix; these elements are then written row by row on a peripheral storage device. The operation is then repeated until all i rows have been rewritten. The matrix may then be said to be "transposed" because the segments of a sequentially stored two-dimensional array are treated by NASTRAN as the columns of a matrix. The number of times that the matrix must be transferred from peripheral storage to high-speed core is T = i/R. The time for data transfer (I/0 time) will be equivalent to that taken to input the full matrix T times and to output it once.

If the matrices are very large, matrix partitioning may be used effectively to reduce the computer time. The matrix is first partitioned by rows and the partitions are then transposed as shown below.

$$[A] = \begin{array}{|c|}\hline A_1 \\ \hline A_2 \\ \hline A_3 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|}\hline & & \\ A_1{}^T & A_2{}^T & A_3{}^T \\ & & \\ \hline \end{array} = [A]^T \quad . \qquad (3)$$

The technique is as follows. The matrix [A] is read into core one column at a time, and the elements in the first P rows of each column are extracted and placed in a peripheral storage file. The operation is repeated, reading the elements in the next P rows by columns into a second peripheral storage file, etc. Thus, since the [A] matrix has i rows, the I/0 time for partitioning is equivalent to that for i/P reads and one write of the complete matrix. Next the $[A_1]$ matrix is

PAGE 2.1-4B INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

transposed by the first-described method which, since $[A_i]$ has P rows, requires P/R reads and one write. Because the columns of $[A_i]^T$ are also columns of $[A]^T$, the transposition is complete when all of the partitions have been transposed. The I/0 time for transposing the partitions is equivalent to P/R reads and one write of the complete [A] matrix. Assuming that reads and writes take the same time, the total time is proportional to the parameter

$$\tau = \frac{i}{P} + 1 + \frac{P}{R} + 1 = T \cdot \frac{R}{P} + \frac{P}{R} + 2 \quad . \tag{4}$$

The number of rows in each partition, P, may be freely selected. The minimum value of the time parameter obtained when $\partial\tau/\partial P = 0$, is

$$\tau_{min} = 2(1 + \sqrt{T}) \quad , \tag{5}$$

and occurs when

$$P = R\sqrt{T} \quad . \tag{6}$$

The time for the second method is less than that for the first when

$$2(1 + \sqrt{T}) < T + 1 \quad , \tag{7}$$

which is satisfied when $T \geq 6$. The second method is automatically selected by the program when this condition is satisfied.

2.1.5 Matrix Partitioning and Merging

In structural analysis, vectors describing the system variables are frequently separated into subsets which are then treated differently. For example, in the displacement method matrix partitioning may be applied to the displacement vector $\{u_f\}$, resulting in two subsets: $\{u_o\}$, degrees of of freedom removed by partitioning, and $\{u_a\}$, degrees of freedom not removed (see Section 3.5.3). All of the arrays associated with $\{u_f\}$, such as the load vector, $\{P_f\}$, and the stiffness matrix, $[K_{ff}]$, must also be partitioned. The partitioning operations are formally indicated as follows:

$$\{P_f\} = \left\{\begin{array}{c} P_a \\ \hdashline P_o \end{array}\right\} \quad , \tag{8}$$

$$[K_{ff}] = \left[\begin{array}{c:c} K_{aa} & K_{ao} \\ \hdashline K_{ao}{}^T & K_{oo} \end{array}\right] \quad . \tag{9}$$

Later in the analysis it will be necessary to recombine the elements of $\{u_o\}$ and $\{u_a\}$. This operation, called the "merge" operation, is formally indicated by

$$\left\{\begin{array}{c} u_a \\ -- \\ u_o \end{array}\right\} \rightarrow \{u_f\} \quad . \tag{10}$$

The essential feature of the operation is that the original order of the members of $\{u_f\}$ must be restored. Order must also be maintained during the partition operation.

The partition and merge operations are accomplished in the program with the aid of USET, an array that describes the membership of each degree of freedom in each of the defined vector sets. There are approximately fifteen such sets (see Section 3.3). One word of USET is assigned to each degree of freedom. One binary bit in each word of USET corresponds to a different vector set. A bit is set equal to unity if the degree of freedom is a member of the corresponding vector set. USET may, consequently, be regarded as a table with marks in appropriate row-column intersections as shown below.

u_g	--	u_f	u_o	u_a	--
✓		✓	✓		
✓		✓	✓		
✓		✓	✓		
✓		✓		✓	
✓		✓	✓		
✓		✓		✓	
✓					
✓		✓	✓		
✓					
✓		✓		✓	
✓					
✓		✓		✓	

In partitioning $[K_{ff}]$ (Equation 9) for example, USET is called into core storage along with the first column of $[K_{ff}]$. USET is scanned and the ordinals of the nonzero bits in the positions corresponding to u_f, u_o, and u_a are noted and copied onto separate lists. The lists are then used to separate the elements in the first (and succeeding) column(s) of $[K_{ff}]$ into $[K_{aa}]$ and $[K_{ao}]^T$, which are then read out of core onto separate files. In the merge operation (Equation 10), the lists are scanned to determine whether a number from $\{u_a\}$ or a number from $\{u_o\}$ will be the next number to be copied into $\{u_f\}$.

2.2 TRIANGULAR DECOMPOSITION

The factoring of a matrix into upper and lower triangular forms is a central feature of structural analysis as performed with the NASTRAN program. For large problems a substantial fraction of the total computing time is associated with triangular decompositions. The NASTRAN program requires a decomposition routine that works well for both full and sparse matrices. Matrices encountered in structural analysis, including structural dynamics, may be either real or complex.

Most of the matrices used in structural analysis are initially very sparse; however, they tend to fill in various degrees as the problem solution proceeds. Under some conditions, matrix multiplications will fill a matrix prior to the beginning of the triangular decomposition. Under other conditions an initially sparse matrix may completely fill during the triangular decomposition. However, for many matrices used in structural analysis, much of the original sparsity is maintained in the triangular factors. In order to handle all of these situations effectively, the decomposition routines treat all matrices as sparse. The procedures efficiently treat the general sparse case as well as the limiting cases of a full matrix or of a simple band matrix.

2.2.1 Triangular Decomposition of Symmetrical Matrices

It is well known (see, for instance, Reference 1) that any square matrix [A], having nonzero leading minors, can be expressed in the form [A] = [L][D][U], where [L] and [U] are unit-lower and unit-upper triangular matrices respectively, and [D] is a diagonal matrix. The matrix [D] can be incorporated entirely within either [L] or [U] or part with each. The different ways of incorporating [D], combined with different orders of operations in determining the terms of [L] and [U], have given rise to many named procedures for performing triangular decompositions.

The following discussion will be based on the equation

$$[A] = [L][U] \quad , \tag{1}$$

where [L] is a unit lower triangle. The elements of the upper triangle may be computed by the following recursion formula:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} \ell_{jk}\, u_{ki} \quad . \tag{2}$$

46

For symmetric matrices without pivoting, the upper and lower triangular elements are related as follows:

$$\ell_{jk} = \frac{u_{kj}}{u_{kk}} \quad . \tag{3}$$

The substitution of the relation in Equation 3 into Equation 2 gives

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} \frac{u_{kj}}{u_{kk}} u_{ki} \quad . \tag{4}$$

Now, $k < i \leq j$, so that only previously computed results occur on the right-hand side of Equation 4 if the elements u_{ij} are computed in order starting with the first row. The unit lower triangle and the associated diagonal elements are saved on secondary storage for later use in equation solution operations.

Figure 1 shows the triangular factor for a sparse matrix. Initial nonzero terms are indicated by X's with O's indicating nonzero terms created as the decomposition proceeds. The terms in triangles indicate the relative locations for nonzero contributions to the upper triangular factor when the first row of the matrix is the pivotal row. If there is sufficient main storage to hold all of the nonzero terms associated with each pivotal row, the decomposition may proceed without the need for writing intermediate results on secondary storage. In general, no nonzero terms will appear in any column of [U] until a nonzero term appears in [A]. The apperance of the first nonzero term in [A] defines the beginning of an "active column." Columns 1, 2 and 9 are active when the first row is the pivotal row. The terms in squares indicate the relative locations for nonzero contributions to the upper triangular factor when the third row of the matrix is the pivotal row. At this point in the decomposition, rows 3, 7, 9 and 13 are active.

If at some point in the decomposition, the diagonal term of the pivotal row initiates a new active column, all existing active columns will terminate in the previous row (change status from active to passive). In row 4 of Figure 1, columns 7, 9 and 13 become passive. Also in row 7, columns 11 and 14 become passive. Prior to using row 4 as a pivotal row, the passive terms in rows 7, 9 and 13 (in squares on Figure 1) are transferred to secondary storage. These terms remain on secondary storage until each of the rows 7, 9 and 13 become pivotal rows, at which time they are transferred to main storage and combined with the original nonzero terms to form each of the pivotal rows. Columns remain passive until a nonzero term appears in that column for a later pivotal row. Nonzero terms which change the status of columns from passive to active may be either in the original matrix, such as column 11 of row 10, or they may be created prior to the column becoming passive, such as columns 9 and 13 of row 7. Figure 1 indicates that these later nonzero terms were created when row 3 was the pivotal row.

If there is sufficient main storage for all of the terms generated by the second term of Equation 4 for each pivotal row, the triangular decomposition can be completed with a single pass through the matrix. When the number of active columns exceeds the capacity of the working storage space, an automatic spill logic is provided. The decomposition proceeds by holding the nonzero terms for as many rows as possible in main storage. Following the completion of all possible pivotal row operations, the intermediate results are transferred to temporary storage. All possible pivotal row operations are then performed on the next group of rows in the matrix, and the intermediate results are transferred to temporary storage. Next, the temporary file is rewound and pivotal operations are continued on the first and second spill groups. This sequence of operations continues, adding the next group of rows on each pass through the matrix, until each pivotal row is complete and transferred to permanent secondary storage.

A preliminary pass is made over the original matrix in order to estimate the execution time and create tables which assist in the efficiency of the decomposition operation. The computing time to perform any calculation may be estimated by counting the number of elementary operations that it involves and assigning experimentally determined values of time to the various types of elementary operations. In the case of triangular decomposition by the method described above, the estimated time is

$$T = \frac{1}{2} M \sum_{i=1}^{N} C_i^2 + I(1+n) \sum_{i=1}^{N} C_i R_i + \frac{1}{2} I \sum C_s^2 + \frac{1}{2} (P_p + P_g) \sum C_t^2 + P_p \sum_{i=1}^{N} C_i \, , \qquad (5)$$

where M = time for multiply-add loop,

I = time to read and write one term on spill file,

P_p = time to put one term in write buffer,

P_g = time to get one term from read buffer,

N = order of matrix,

C_i = number of active columns in the ith row,

R_i = number of I/O transfers for the ith row. R_i may be approximated by the integral part of C_i/S, where S = number of core-held rows in the current spill group,

C_s = number of active columns at beginning of spill operations that are out of range of first spill group (column numbers greater than last row in spill group) for each time that spill operations begin,

C_t = sum of number of passive columns on secondary storage and number of active columns in working space for each time that active column termination occurs, and

n = number of words per term.

The computing time is dominated by the first term in Equation 5, which is associated with the arithmetic operations in the step-by-step elimination procedure. Since the number of active columns is a function of the ordering of the matrix, the user can shorten the computing time by ordering the matrix in the most favorable manner. A discussion of the sequencing of grid points to minimize the time required for triangular decomposition is given in Section 1 of the User's Manual.

The second and third terms of Equation 5 are zero, unless spill operations require the transfer of intermediate results to secondary storage. The fourth term is the modest overhead associated with passive columns, and the last term is the time required to transfer the final result of the triangular decomposition to secondary storage.

In order to assist the user in locating singularities, or near singularities in the matrix, information relative to the magnitude of the diagonal elements of the triangular factor is furnished to the user. The absolute value (ε) of the ratio of the diagonal element in the original matrix to the diagonal element in the triangular factor is determined for each row of the matrix. The maximum value of ε along with the distribution of the values of ε is furnished as diagnostic information. The row numbers for the five largest values of ε are also furnished along with the number of negative values for the diagonal element in the triangular factor.

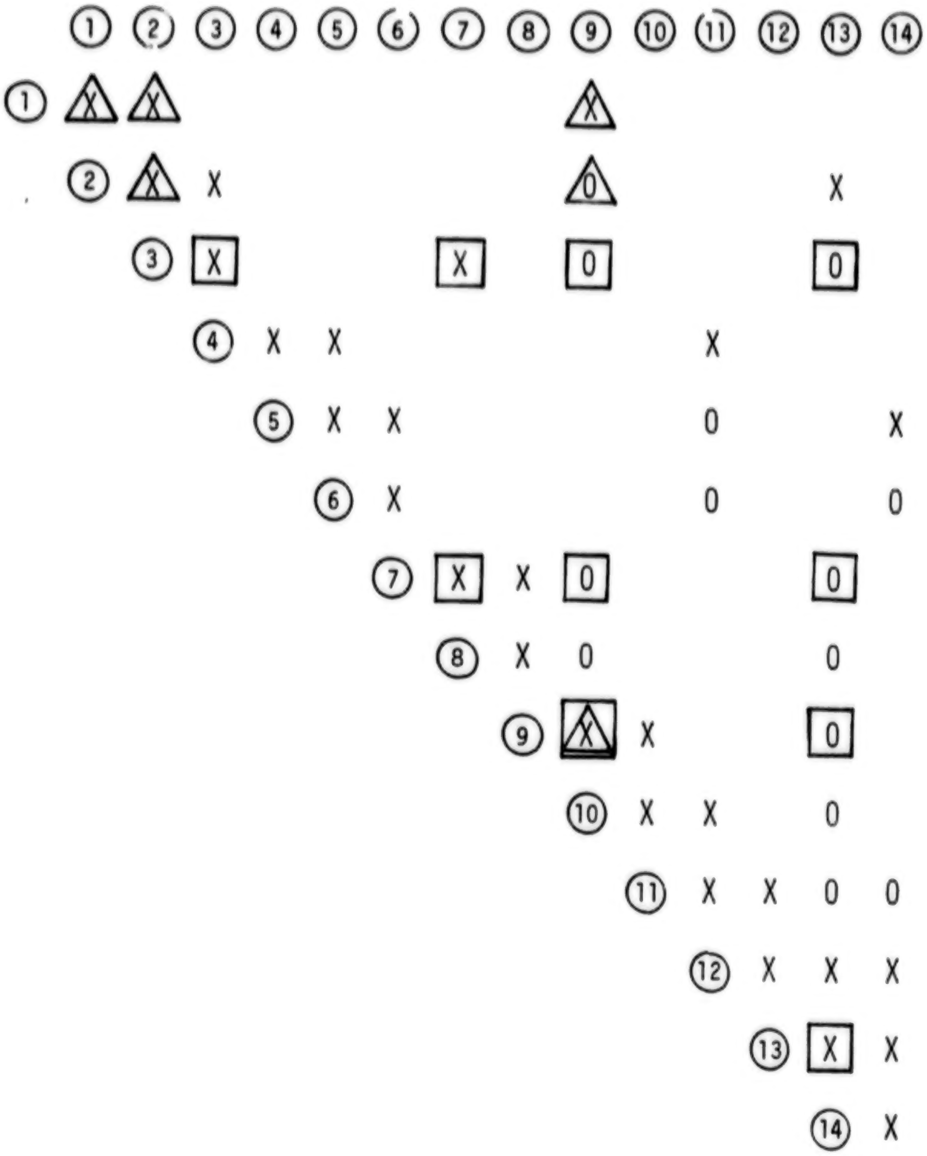


Figure 1. Triangular factor for sparse matrix

2.2.2 Triangular Decomposition of Unsymmetrical Matrices

The procedures for the triangular decomposition of unsymmetrical partially banded matrices are similar to those used for symmetrical matrices. The lack of symmetry means that the upper and lower triangular factors are not related, and that the widths of the upper and lower bands (which replace the symmetrical semi-bands of the symmetrical matrix) may be different. However, the band structure of the original matrix will be maintained in the triangular factors.

Although the lack of symmetry means that the pattern of scattered terms outside the upper band may be different than the pattern outside the lower band, it still remains true that no nonzero terms will appear in any column of the upper triangular factor until a nonzero term appears in the same column of the original matrix. Likewise, no nonzero terms will appear in any row of the lower triangular factor until a nonzero term appears in the same row of the original matrix. Hence the partially banded nature of the matrix is maintained after the completion of the triangular decompositon.

The lack of any assurance that all leading minors are nonsingular requires that pivoting (i.e., interchange of rows) be used to maintain the numerical stability of the triangular decomposition. Pivoting is restricted to take place within the lower band. This will increase the bandwidth of the upper triangular factor by the width of the lower band, but will not otherwise affect the partially banded character of the triangular factors.

The general procedure for an unsymmetrical decomposition will be discussed with reference to Figure 2, which shows an unsymmetrical partially banded matrix of order N, upper bandwidth B, and lower bandwidth $\bar{B}$, with several nonzero terms outside the bands. Initial nonzero terms are indicated by x's, with O's indicating nonzero terms created outside the original bands as the decomposition proceeds. The O's within the expanded upper band $\bar{B}$ indicate the maximum number of nonzero terms that can be created by the pivoting. The existence of initial zero terms inside the lower band $\bar{B}$ and the expanded upper band $B + \bar{B}$ is ignored as, in general, these terms will become nonzero as the decomposition proceeds.

If there is sufficient core storage to hold $B + \bar{B}$ columns of the lower triangular factor, as indicated inside the solid parallelogram of Figure 2, along with the associated active column and active row terms, the triangular decomposition can be completed with a single pass through the matrix. Otherwise secondary storage must be used for intermediate results and provision is made in main storage to hold R columns of the lower triangular factor, a single column of the upper triangular factor, and the current active columns and active rows.

The decomposition begins by reading the original matrix one column at a time, pivoting the largest term in absolute value within the lower band to the diagonal position, and determining the inner products for the current column, including the active row terms. The portion of the column in the lower triangular factor, including active row terms, is retained in working storage. The portion of the column in the upper triangular factor within the expanded upper band is complete and no longer needed; hence it can be written on a secondary storage device. This continues until R columns have been processed. At this point the procedure is changed only to the extent that the portion of the current column within the lower band is temporarily stored on a secondary device.

The decomposition continues until $B + \overline{B}$ columns have been processed. At this point, the first column of the lower triangular factor, including the active row terms, is no longer needed and can be written on a secondary storage device. This releases $\overline{B}$ spaces in working storage. This procedure continues until the decomposition is completed.

The active column terms are transposed prior to beginning the decomposition, so they are available by rows and can be read into main storage as needed. If an active column term exists in the i^{th} row, it is stored along with the $i + B$ column of the upper triangular factor.

A preliminary pass is made over the original matrix in order to locate the extreme nonzero terms for each row in the lower triangle and each column in the upper triangle. The maximum number of active columns is determined by counting the maximum number of intersections for any row with columns defined by drawing lines from the most extreme nonzero term in the upper triangle to the outside edge of the upper band. The maximum number of active rows is determined by counting the maximum number of intersections for any column with rows defined by drawing lines from the most extreme nonzero term in the lower triangle to the outside edge of the upper band. An examination of the matrix shown in Figure 2 reveals that the maximum number of active columns is 2 even though the total number of nonzero columns outside the upper band is 3. The lower triangle contains 3 active rows and 4 nonzero rows outside the lower band.

As with the symmetrical decomposition, the routine selects the bandwidths that give the minimum computing time based on the ordering of the matrix presented. Proper sequencing is similar to that used for symmetrical matrices.

The computing time will again be a function of the working storage available for the execution of the routine. Working storage consists of space for R columns of terms inside the band for the lower triangular factor, $\bar{B} + B$ spaces for the current column of the upper triangular factor, $\bar{B}C$ spaces for active column terms, $(\bar{B} + B)\bar{C}$ spaces for active row terms, $C\bar{C}$ spaces for interaction of active row and active column terms, and $B + \bar{B}$ spaces for the permutation matrix. This results in working storage as follows:

$$W = \bar{B}R + 2\bar{B} + 2B + C\bar{B} + \bar{C}(\bar{B}+B) + C\bar{C} , \qquad (14)$$

where $R \geq 1$, $R \leq B + \bar{B}$, $R \leq N$, C is the maximum number of active columns, and $\bar{C}$ is the maximum number of active rows.

The computing time to perform an unsymmetrical triangular decomposition is:

$$T = T_1 + T_2 + T_3 + T_4 , \qquad (15)$$

where T_1 is the time required to process the first $N - B - 2\bar{B}$ columns of terms inside the upper and lower bands, T_3 is the time required to process the last $\bar{B}$ columns of terms inside the bands, and T_2 is the time required to process the remaining intermediate $B + \bar{B}$ columns of terms inside the bands. T_4 is the time required to process the active row and active column terms.

$$T_1 = K_1[M_B\bar{B}R + I\bar{B}(B + \bar{B} - R) + P(B + 2\bar{B})] , \qquad (16)$$

where M_B is the arithmetic time required to process one term inside the bands, I is the time required to store and retrieve one term inside the lower band, and P is the time required to store one term of the final result on a secondary storage device. If $N > B + 2\bar{B}$, then $K_1 = N - B - 2\bar{B}$. If $N \leq B + 2\bar{B}$, then $K_1 = 0$.

$$T_2 = \frac{K_2}{2}[\bar{B}K_2M_B + (K_3 - R)(I - M_B)\bar{B} + 2P\bar{B} + PK_2] . \qquad (17)$$

If $N \geq B + 2\bar{B}$, then $K_2 = K_3 = B + \bar{B}$. If $N < B + 2\bar{B}$, then $K_2 = N - \bar{B}$ and $K_3 = B + \bar{B}$, unless $N < B + \bar{B}$, then $K_3 = N$.

$$T_3 = \frac{\bar{B}^3}{3} M_B + \frac{K_4^3}{2} I + P\bar{B}K_5 . \qquad (18)$$

If $N \geq B + 2\bar{B}$, then $K_4 = B + \bar{B} - R$ and $K_5 = B + \frac{3}{2}\bar{B}$, unless $B > R$, then $K_4 = \bar{B}$. If $N < B + 2\bar{B}$, then $K_4 = N - R$ and $K_5 = N$, unless $N - R > \bar{B}$, then $K_4 = \bar{B}$.

$$T_4 = (N - \bar{B})[M_C(\bar{B}C + B\bar{C} + \bar{B}\bar{C} + C\bar{C}) + P(C + \bar{C})], \qquad (19)$$

where M_C is the arithmetic time required to process one active row or active column term.

If N is assumed large compared to both B and $\overline{B}$ and the final storage terms are neglected, Equation 15 can be simplified as follows:

$$T = N[M_B\overline{B}R + M_C(\overline{B}C + \overline{B}\overline{C} + B\overline{C} + C\overline{C}) + I\overline{B}(B + \overline{B} - R)]. \qquad (20)$$

This simplified equation is used for making timing calculations in selecting the optimum band widths and active elements.

The sequence of events in selecting the bandwidths and active elements outside the bands may be summarized as follows:

1. Locate extreme nonzero terms in each column for the upper triangle and in each row of the lower triangle.
2. Prepare a table of unique pairs of upper bands and active columns.
3. For the working storage available, compute R using Equation 14.
4. Assuming $B = \overline{B}$ and $C = \overline{C}$, and using Equation 20 determine the upper bandwidth and the associated number of active columns that result in minimum computer time to perform the triangular decomposition.
5. Using the previously determined upper band and active columns, determine the lower bandwidth and the associated number of active rows that result in minimum computer time to perform the triangular decomposition according to Equations 14 and 20.
6. Select the values of the bandwidths and active elements that result in minimum time to perform the trangular decomposition and recalculate the time using Equation 15. This more accurate time estimate is needed because decisions are made by modules using the decomposition routines that are based on the estimated running time.

The complex decomposition routine is the same as the real unsymmetric routine, except that twice as much storage is needed for complex numbers and the real arithmetic is replaced with complex arithmetic.

MATRIX OPERATIONS

REFERENCE

1. Forsythe, G., and Moler, C. B., Computer Solution of Linear Algebraic Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., p, 27.

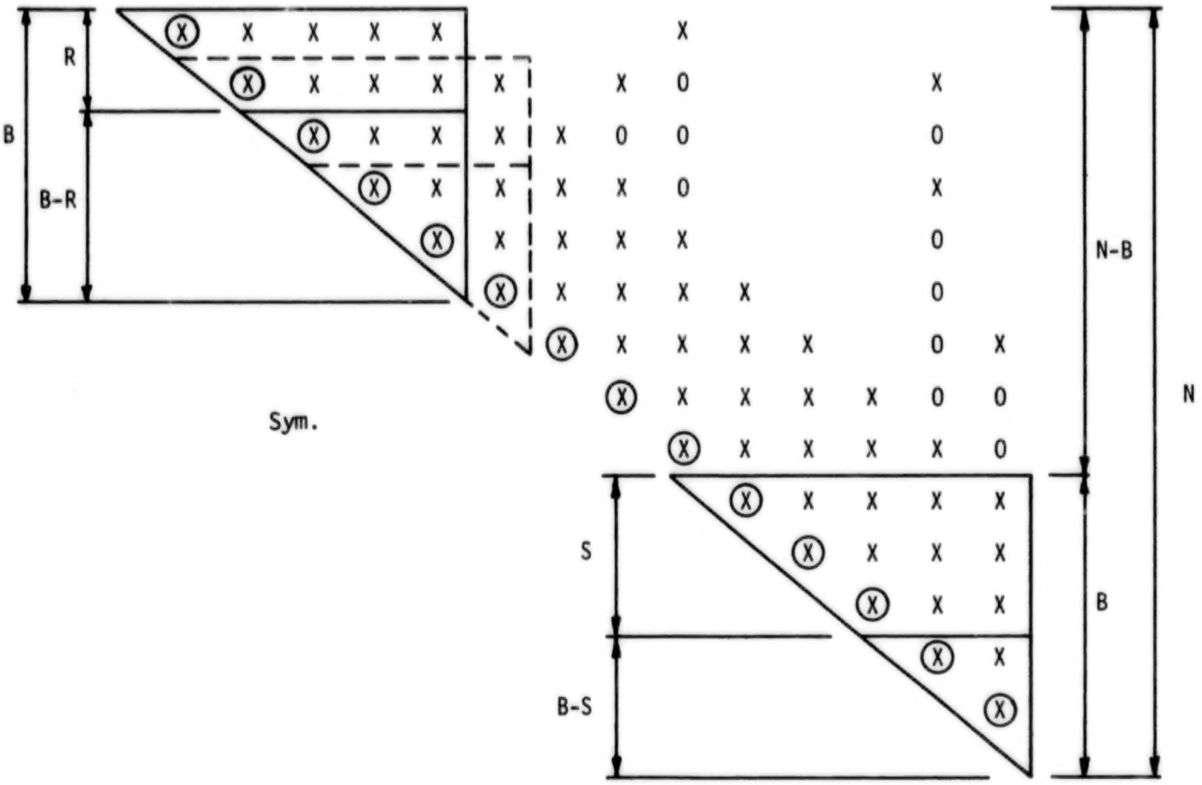

B
R
B-R
Sym.
S
B-S
N-B
B
N

Figure 1. Symmetrical partially banded matrix.

56

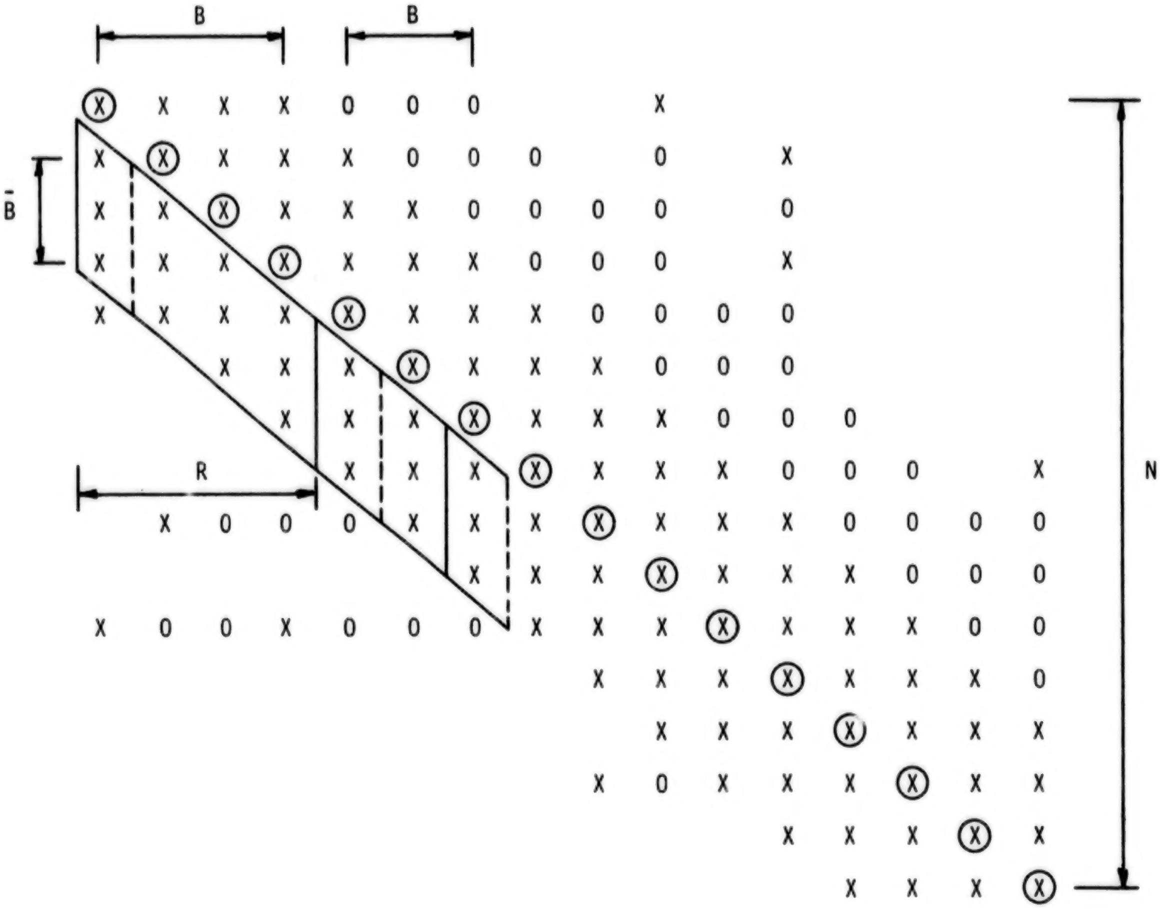


Figure 2. Unsymmetrical partially banded matrix.

2.3 SOLUTION OF [A]{x} = {b}

The solution of the equation

$$[A]\{x\} = \{b\} \quad , \tag{1}$$

is accomplished using the results of the decomposition procedure described in Section 2.2. Replacing [A] by its triangular factors, Equation 1 becomes

$$[L][U]\{x\} = \{b\} \quad . \tag{2}$$

where [L] is a lower unit triangle and [U] is an upper triangle.

Define

$$\{y\} = [U]\{x\} \quad . \tag{3}$$

Then, substituting into Equation 2,

$$[L]\{y\} = \{b\} \quad . \tag{4}$$

The solution of Equation 4 for {y} is called the forward pass, and the subsequent solution of Equation 3 for {x} is called the backward pass.

In the solution algorithm, y_1 is evaluated from the leading element of [L], and the nonzero elements in the first column of [L] are multiplied by y_1 and transferred to the right hand side of Equation 4. The procedure is repeated for the second and succeeding columns of [L] until all elements of {y} have been evaluated. The algorithm for obtaining {x} is similar except that the columns of [U] are required in reverse order. Multiple {b} vectors can be handled simultaneously up to the limit of the working space available in main memory. The same general procedures are used for both symmetric and unsymmetric matrices.

The forward pass requires the reading of both the right hand vectors and the lower triangular factor from secondary storage devices. In the case of symmetric matrices, the processor time associated with the location of the terms in the lower triangular factor is minimized by working directly in the I/0 buffers. Also, in the case of symmetric matrices, successive values of {y} are tested for zero prior to the multiplication. In this manner full advantage is taken of the sparsity of the right hand side on the forward pass.

For symmetric matrices, a term in the upper triangular factor is easily obtained from the corresponding term in the transposed location of the lower triangular factor (see Equation 3 of Section 2.2.1). Thus, the backward pass utilizes a special packing format which allows for the backward reading of the lower triangular factor. For unsymmetric operations, the backward pass is accomplished in two steps. First, the upper triangular factor is read backward and written forward on a separate file so that the last column of [U] appears first. This is part of the triangular decomposition routine and takes place immediately after the completion of the decomposition. The second step consists of solving Equation 3 for {x}. It is made part of the equation solution routine.

Following the determination of the solution vectors, a residual vector is determined for each solution vector as follows:

$$\{\delta b\} = \{b\} - [A]\{x\} \quad . \tag{4}$$

The residual vector is used to calculate the following error ratio which is printed with the output.

$$\varepsilon = \frac{\{x\}^T\{\delta b\}}{\{x\}^T\{b\}} \quad . \tag{5}$$

The magnitude of this error ratio gives an indication of the numerical accuracy of the solution vectors. The computer time required to calculate this error ratio is only a small fraction of the time required to determine the solution vector.

3. STATIC ANALYSIS BY THE DISPLACEMENT METHOD

3.1 INTRODUCTION

From a theoretical viewpoint, the formulation of a static structural problem for solution by the displacement method is completely described by the matrix equation

$$[K]\{u\} = \{P\}. \tag{1}$$

As a matter of practical calculation, there is rather more to the problem than this simple formula would imply, since it is necessary to generate the stiffness matrix [K] and the load vector {P} from the available information about the structure, and to calculate stresses and other quantities of interest from the independent displacement vector, {u}. In the early days of computer-aided analysis these tasks were left to the analyst and the computer busied itself with obtaining the solution to Equation 1. It was soon discovered that, for most practical problems, the computer had only partly unburdened the user and that larger savings of time and cost could be achieved if the computer took over the major share of input data preparation and output data processing. Automatic performance of these additional tasks requires that a particular approach to structural analysis be selected and incorporated into the program.

NASTRAN embodies a lumped element approach, i.e., the distributed physical properties of a structure are represented by a model consisting of a finite number of idealized substructures or elements that are interconnected at a finite number of points. All input and output data pertain to the idealized structural model.

The idealized structural model in NASTRAN consists of "grid points " (G) to which "loads" (P) are applied, and at which degrees of freedom are defined, and "elements" (E) that are connected between the points, as shown in Figure 1. Two general types of grid points are employed in static analysis. They are:

1. Geometric grid point - a point in three-dimensional space at which three components of displacement and three components of rotation are defined. The coordinates of each grid point are specified by the user. Components of displacement and rotation can be eliminated as degrees of freedom by means of "single-point constraints".

2. Scalar point - a point in vector space at which one degree of freedom is defined. A geometric grid point contains from one to six scalar points. Scalar points may exist that are not associated with grid points. Such points can be coupled to geometric grid points by means of scalar structural elements and by constraint relationships.

The structural element is a convenient localizing concept for specifying many of the properties of the structure, including material properties, mass distribution and some types of applied loads. In static analysis by the displacement method, stiffness properties are input exclusively by means of structural elements. Mass properties (used in the generation of gravity and inertia loads) are input either as properties of structural elements or as properties of grid points. In dynamic analysis mass, damping, and stiffness properties may be input either as the properties of structural elements or as the properties of grid points.

The structural elements are described in detail in Section 5 of the Theoretical Manual. There are four general classes of structural elements as follows:

1. Metric elements which are connected between geometric grid points. Examples include rod, plate and shell elements.
2. Scalar (or zero-dimensional) elements which are connected between pairs of scalar points, or between one scalar point and "ground". Note that, since each geometric grid point contains a number of scalar points corresponding to specific components of motion, scalar elements can be connected between selected components of motion at geometric grid points.
3. General elements, whose properties are defined in terms of deflection influence coefficients (i.e., compliance matrices), and which may be interconnected between any number of geometric and scalar grid points. An important application of general elements is the representation of large pieces of structure by means of test data.
4. Constraint Elements (or Constraints). The existence of a constraint element implies a linear relationship among the degrees of freedom to which it is attached of the form

$$\sum_g R_{cg} u_g = Y_c \quad , \tag{2}$$

where u_g are degrees of freedom and Y_c is an enforced displacement. A linear relationship among the forces of constraint is also implied, since it is required that the forces of constraint do no work.

Constraint elements are employed for the following purposes:

a. To introduce enforced displacements.

b. To enforce zero motion in specified directions at points of reaction.

c. To simulate very stiff (rigid) structural members.

d. To describe part of a structure by experimentally determined vibration modes. (The matrix of eigenvectors expresses a relationship of constraint between physical and modal coordinates.)

e. To generate nonstandard structural elements by combining scalar structural elements and constraints.

The constraint concept is important for the displacement method in order to eliminate ill-conditioning generated by very stiff members. Two types of constraint elements are provided: single-point constraints, wherein Equation 2 includes only a single term on the left hand side; and multipoint constraints and rigid elements wherein Equation 2 includes more than one term. The main reason for the distinction is that due to the simplicity of single-point constraints, they are processed separately in the program.

Solution of a linear static structural problem by the displacement method requires a set of preliminary operations which reduce the input data to the matrix form given in Equation 1. Among these operations are the elimination of displacement components that are declared to be dependent by virtue of constraints and the transfer of all applied loads to the independent displacement components.

As input data in static analysis, the loads are specified in a variety of ways including:

1. Concentrated loads at geometric and scalar grid points.
2. Pressure loads on two-dimensional structural elements.
3. Indirectly, by means of the mass and thermal expansion properties of structural elements.

Enforced deformations are also reduced to a set of equivalent loads on the independent displacement components. See Section 3.6.1.

Once Equation 1 has been formed it is solved for each specific loading condition. Stresses in the structural elements and other desired results are then obtained from $\{u\}$ by a set of data recovery operations.

62

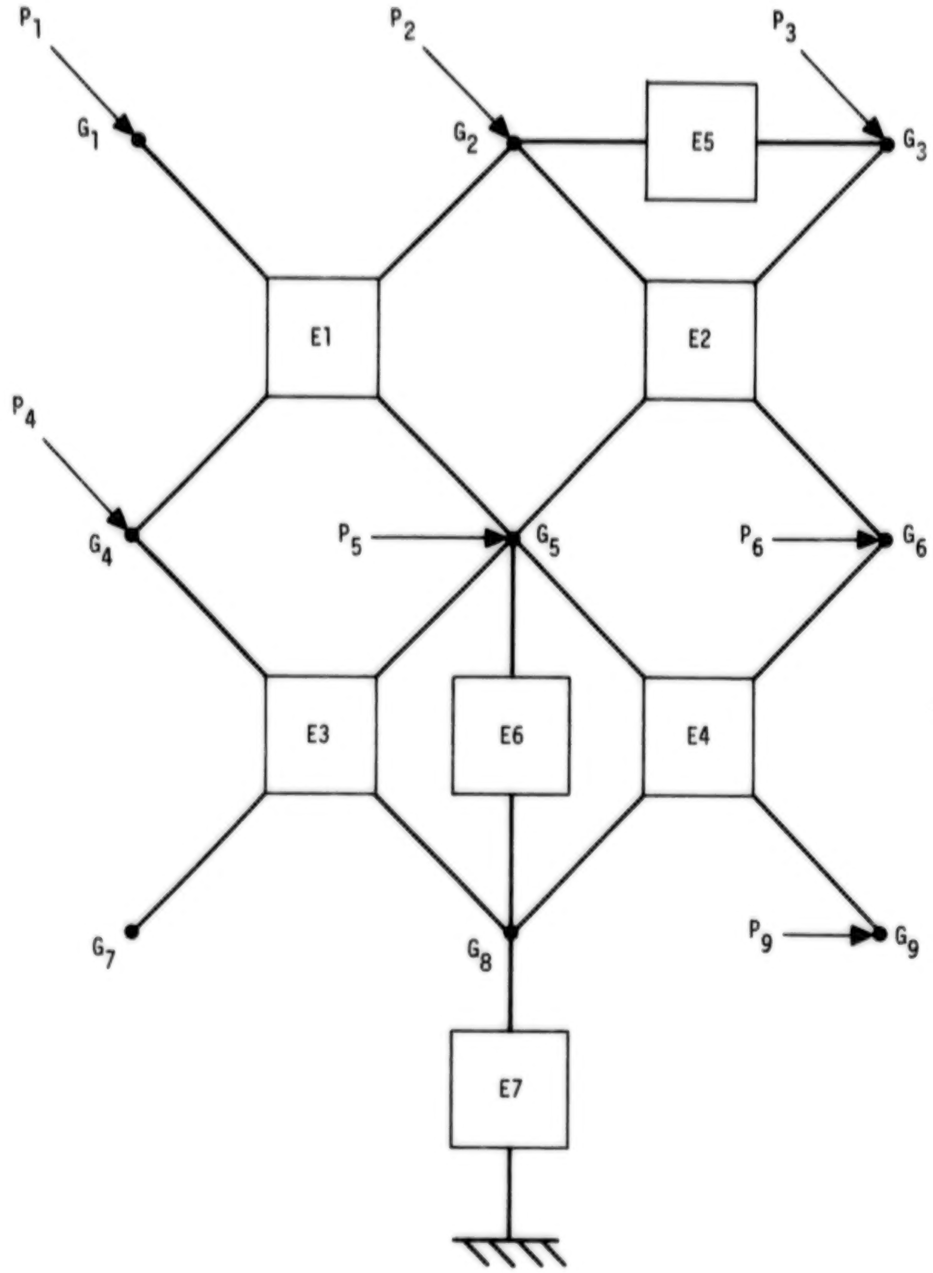

P_1
P_2
P_3
G_1
G_2
E5
G_3
E1
E2
P_4
G_4
P_5
G_5
P_6
G_6
E3
E6
E4
G_7
G_8
P_9
G_9
E7

Figure 1. Topology of the idealized structural model.

3.2 GENERAL PROBLEM FLOW

As has been explained in Section 1 (Program Organization), NASTRAN consists of a number of subprograms, or modules, that are executed according to a sequence of macro-instructions that is controlled by the Executive System. A number of such sequences, called Rigid Formats, are permanently stored in the program and can be selected by means of control cards. Each rigid format corresponds to a particular type of structural analysis. The user may, in addition, devise his own sequence of module calls (referred to as a DMAP sequence) for problems that do not conform to one of the available rigid formats.

The following rigid formats are currently available for the solution of static problems by the displacement method:

1. (Basic) Static Analysis
2. Static Analysis with Inertia Relief
4. Static Analysis with Differential Stiffness
5. Buckling
6. Piecewise Linear Analysis
14. Static Analysis using Cyclic Symmetry

Figure 1 shows a simplified flow diagram for Basic Static Analysis. Each block in the flow diagram represents a number of program modules. The actual number of modules called is approximately equal to thirty. The functions indicated in Figure 1 are described in succeeding subsections of the Theoretical Manual. It suffices at present to indicate the general nature of the tasks performed.

The Input File Processor, as the name implies, reorganizes the information on input data cards into Data Blocks consisting of lists of similar quantities.

The Geometry Processor generates coordinate system transformation matrices, tables of grid point locations, a table defining the structural elements connected to each grid point, and other miscellaneous tables such as those defining static loads and temperatures at grid points.

The Structure Plotter generates tape output for an automatic plotter that will plot the structure (i.e., the location of grid points and the boundaries of elements) in one of several available three-dimensional projections. The structure plotter is particularly useful for the detection of errors in grid point coordinates and in the connection of elements to grid points.

64

Note that the structures plotter may also be used at the end of the program to superimpose images of the deformed and undeformed structure.

The Structural Matrix Assembler generates stiffness and mass matrices referred to the grid points from tabular information generated by the Input File Processor and the Geometry Processor. The mass matrix is used in static analysis for the generation of gravity loads and inertia loads on unsupported structures.

In block 5 of Figure 1, the stiffness matrix is reduced to the form in which it is finally solved through the imposition of single and multi-point constraints, and the use of matrix partitioning (optional).

Load vectors are then generated from a variety of sources (concentrated loads at grid points, pressure loads on surfaces, gravity, temperature, and enforced deformations) and are reduced to final form by the application of constraints and matrix partitioning.

The solution for independent displacements is accomplished in two steps: Decomposition of the stiffness matrix [K] into upper and lower triangular factors; and solution for {u} for specific load vectors, {P}, by means of successive substitution into the equations represented by the triangular factors of [K] (the so-called forward and backward passes). All load vectors are processed before proceeding to the next functional block.

In block 8 of Figure 1, dependent displacements are determined from the independent displacements by means of the equations of constraint. The internal forces and stresses in each element are then computed from knowledge of the displacement components at the corners of the elements and the intrinsic structural equations of the element. Finally the Output File Processor prepares the results of the analysis for printing.

The Loop for Additional Constraint Sets shown in Figure 1, is introduced to facilitate solutions for different boundary conditions, which are applied by means of single point constraints. In particular, the symmetric and antisymmetric responses of a symmetric structure are treated in this manner.

The flow diagram for Rigid Format No. 2, Static Analysis with Inertia Relief, is, to the level of detail considered here, identical to Figure 1. The inertia relief effect consists of a modification to the load vector to include inertia loads due to the acceleration of an unrestrained structure. The manner in which the incremental load is calculated is explained in

65

Section 3.6.3

A simplified flow diagram for Rigid Format No. 4, Static Analysis with Differential Stiffness, is shown in Figure 2. A comparison between Figures 1 and 2 shows that the first eight blocks are identical.

Contributions to the differential stiffness matrix are not defined for all elements currently in NASTRAN, and they may not be defined for a new element. The differential stiffness matrix, which is a first order approximation to large deformation effects, is a function of the most recently iterated displacement. Functional Module, DSCHK, (block 23 of Figure 2) performs differential stiffness calculations based on user-supplied iteration parameters. The solution strategy basically involves a load adjustment (the "inner" loop) in order to satisfy iterated displacements within a specified convergence criterion.

A simplified flow diagram for Rigid Format No. 5, Buckling, is shown in Figure 3. In it the differential stiffness matrix $[K^d]$ corresponding to a particular applied loading condition is used in conjunction with the structural stiffness matrix [K] to formulate an eigenvalue problem

$$[K + \lambda K^d]\{u\} = 0. \qquad (1)$$

The eigenvalues, λ_i, are the load level factors for various buckling modes. They and the corresponding eigenvectors, $\{\phi_i\}$, are extracted by the Real Eigenvalue Analysis module. Additional data (constrained displacement components and stress patterns for each buckling mode) are recovered in Block 15, which is virtually a duplicate of Block 8, and the buckling mode shapes are plotted, if desired.

A simplified flow diagram for Rigid Format No. 6, Piecewise Linear Analysis, is shown in Figure 4. In piecewise linear analysis solutions are obtained for structures with nonlinear, stress-dependent, material properties. The load level is increased to its full value by small increments, such that stiffness properties can be assumed to be constant over each increment. After each increment, the combined strains in nonlinear elements due to all load increments are used, in conjunction with stress-strain diagrams, to determine the appropriate stiffnesses for the next load increment. The procedures, summarized in Figure 4, are described in Section 3.8.

ORIGINAL PAGE IS
OF POOR QUALITY

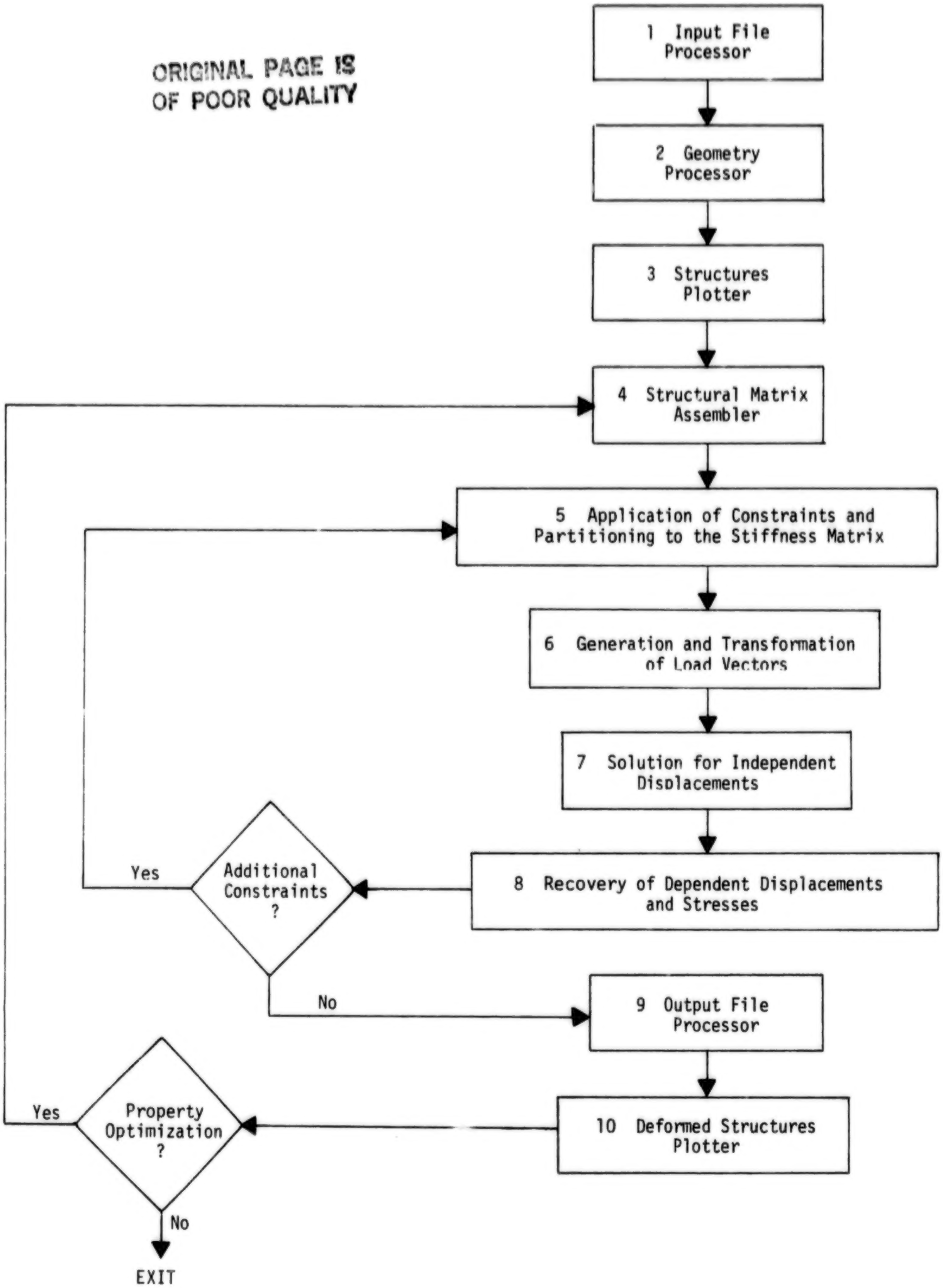

1 Input File Processor
2 Geometry Processor
3 Structures Plotter
4 Structural Matrix Assembler
5 Application of Constraints and Partitioning to the Stiffness Matrix
6 Generation and Transformation of Load Vectors
7 Solution for Independent Displacements
8 Recovery of Dependent Displacements and Stresses
Additional Constraints ?
Yes
No
9 Output File Processor
10 Deformed Structures Plotter
Property Optimization ?
Yes
No
EXIT

Figure 1. Simplified flow diagram for Basic Static Analysis.

61

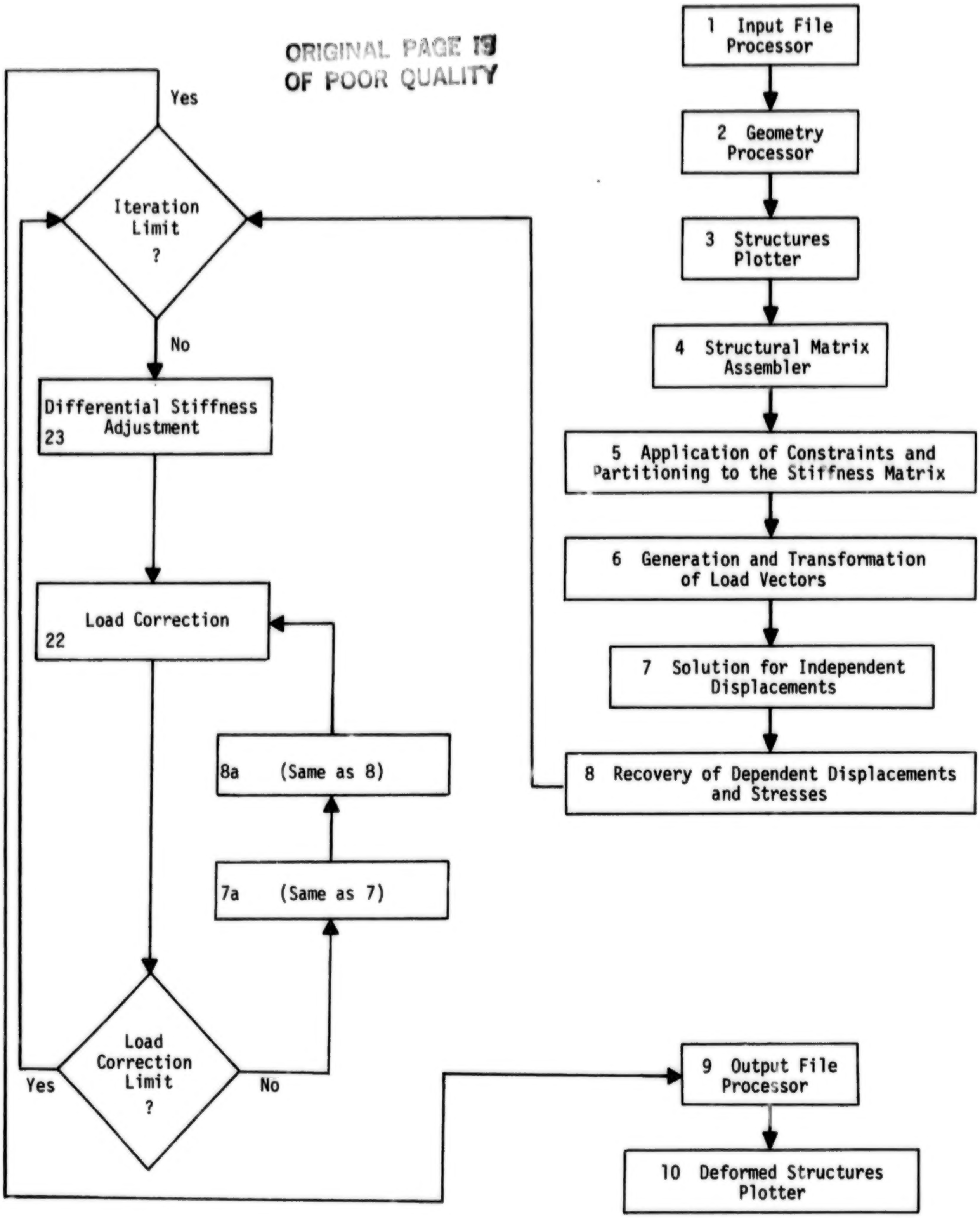

1 Input File Processor
2 Geometry Processor
3 Structures Plotter
4 Structural Matrix Assembler
5 Application of Constraints and Partitioning to the Stiffness Matrix
6 Generation and Transformation of Load Vectors
7 Solution for Independent Displacements
8 Recovery of Dependent Displacements and Stresses
Iteration Limit ?
Yes
No
Differential Stiffness Adjustment
23
Load Correction
22
8a (Same as 8)
7a (Same as 7)
Load Correction Limit ?
Yes
No
9 Output File Processor
10 Deformed Structures Plotter

Figure 2. Simplified flow diagram for Static Analysis with Differential Stiffness.

STATIC ANALYSIS BY THE DISPLACEMENT METHOD

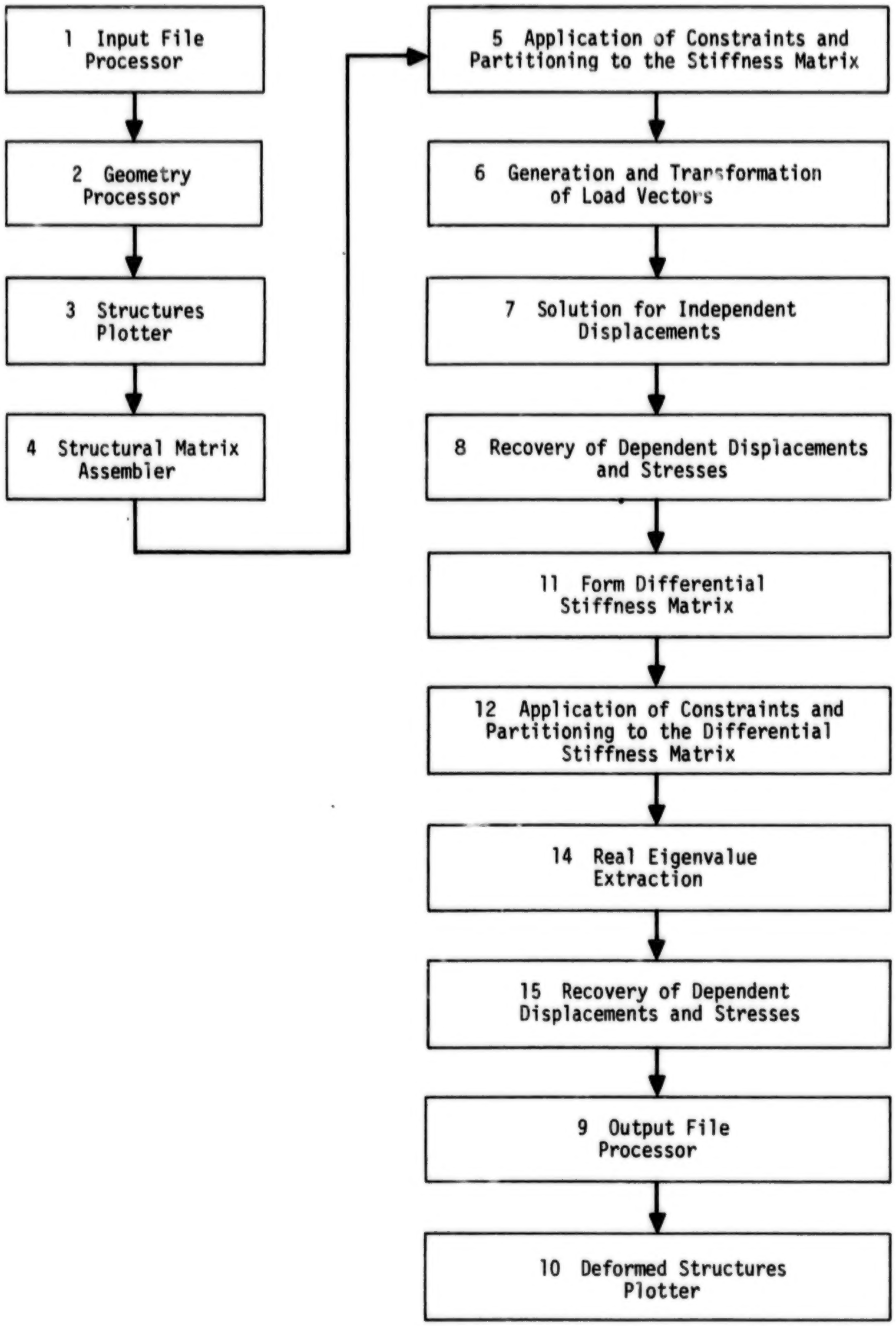

1 Input File Processor
2 Geometry Processor
3 Structures Plotter
4 Structural Matrix Assembler
5 Application of Constraints and Partitioning to the Stiffness Matrix
6 Generation and Transformation of Load Vectors
7 Solution for Independent Displacements
8 Recovery of Dependent Displacements and Stresses
11 Form Differential Stiffness Matrix
12 Application of Constraints and Partitioning to the Differential Stiffness Matrix
14 Real Eigenvalue Extraction
15 Recovery of Dependent Displacements and Stresses
9 Output File Processor
10 Deformed Structures Plotter

Figure 3. Simplified flow diagram for buckling.

GENERAL PROBLEM FLOW

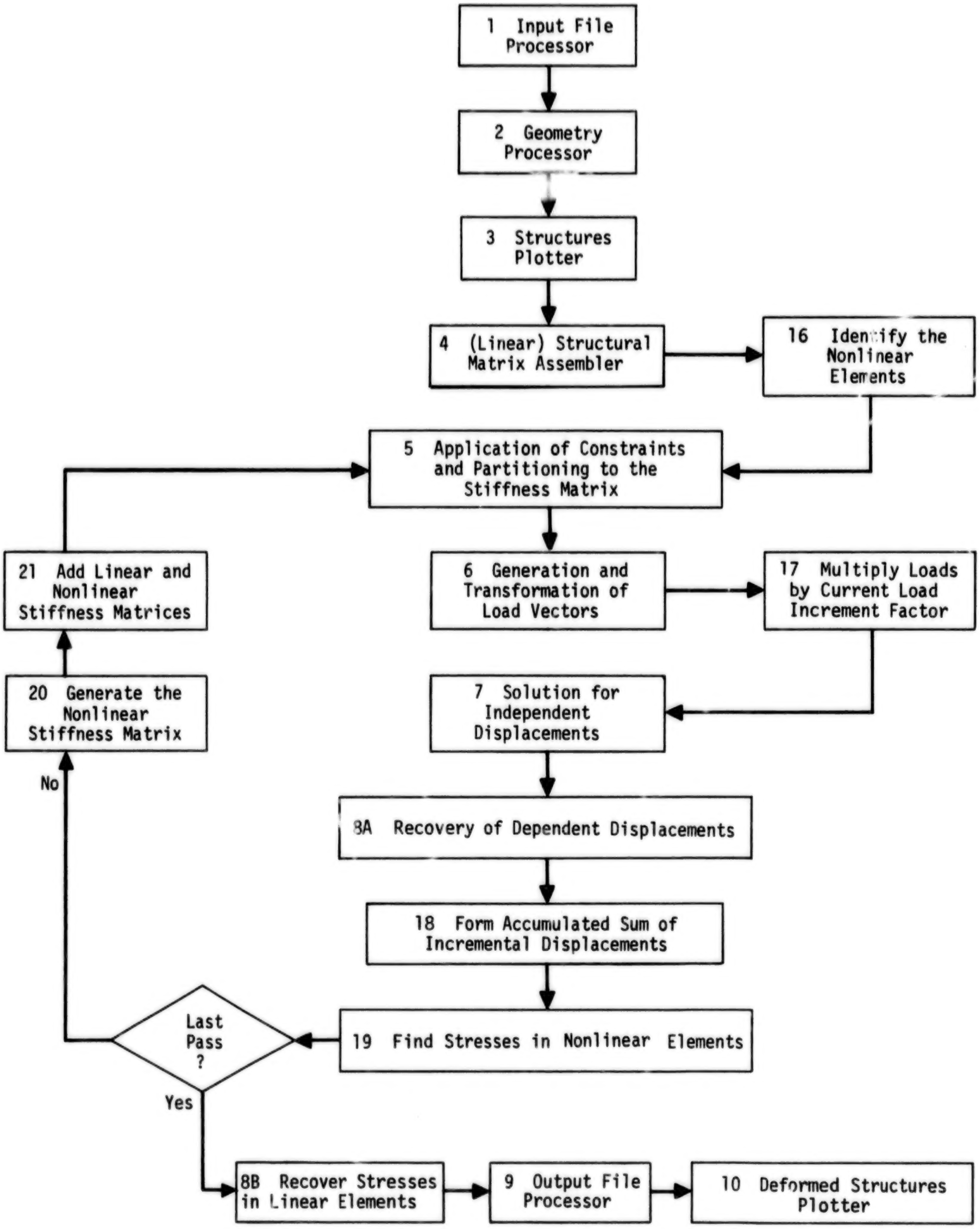

1 Input File Processor
2 Geometry Processor
3 Structures Plotter
4 (Linear) Structural Matrix Assembler
16 Identify the Nonlinear Elements
5 Application of Constraints and Partitioning to the Stiffness Matrix
21 Add Linear and Nonlinear Stiffness Matrices
6 Generation and Transformation of Load Vectors
17 Multiply Loads by Current Load Increment Factor
20 Generate the Nonlinear Stiffness Matrix
7 Solution for Independent Displacements
No
8A Recovery of Dependent Displacements
18 Form Accumulated Sum of Incremental Displacements
Last Pass ?
19 Find Stresses in Nonlinear Elements
Yes
8B Recover Stresses in Linear Elements
9 Output File Processor
10 Deformed Structures Plotter

Figure 4. Simplified flow diagram for piecewise linear analysis.

3.3 NOTATION SYSTEM

Many of the operations performed in computerized structural analysis are conveniently expressed in the notation of matrix algebra. In NASTRAN matrix arrays are represented by a root symbol that indicates the type of physical quantity and by one or more subscripts and superscripts that act as modifiers. The root symbols used in static analysis by the displacement method are listed in Table 1. Square brackets, [ ], indicate two-dimensional arrays and twisted brackets, { }, indicate column vectors. Row vectors, which are less common, are usually indicated by appending the transpose symbol, T, to the twisted brackets.

Subscripts are used exclusively to designate the subsets of displacement components to which the root symbol applies as for example in the equation,

$$\{q_s\} = -\{P_s\} + [K_{fs}]^T\{u_f\} + [K_{ss}]\{u_s\}, \qquad (1)$$

which is used to recover single point forces of constraint, $\{q_s\}$, from displacements at constrained points, $\{u_s\}$, and at unconstrained (free) points, $\{u_f\}$. Nearly all of the matrix operations in static analysis are concerned with partitioning, merging and transforming matrix arrays from one subset of displacement components to another. All the components of displacement of a given type (such as all points constrained by single-point constraints) form a vector set that is distinguished by a subscript from other sets. A given component of displacement can belong to several vector sets. The mutually exclusive vector sets, the sum of whose members are the set of all physical components of displacement, $\{u_p\}$, are listed in Table 2a.

In addition, a number of vector sets are defined as the union of two or more independent sets. See Table 2b.

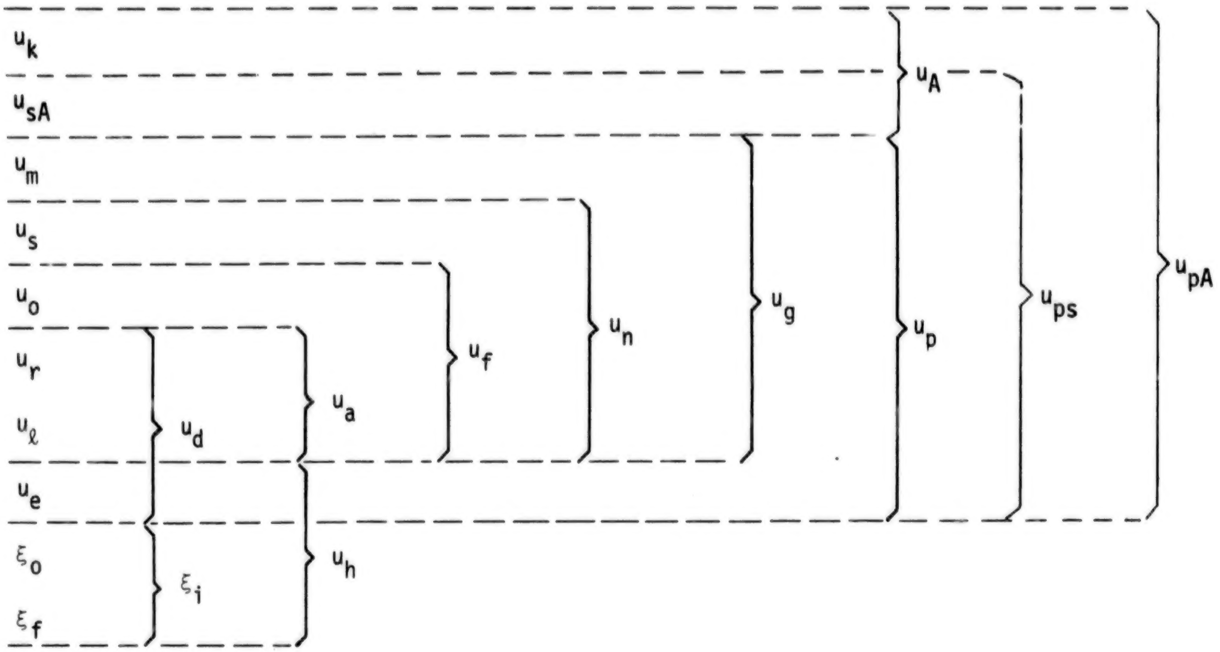
In dynamic analysis, additional vector sets are obtained by a modal transformation derived from real eigenvalue analysis of the set $\{u_a\}$. See Table 2c.

In aeroelastic analysis, additional vector sets are defined by the aerodynamic degrees of freedom. See Table 2d.

The nesting of the vector sets in Table 2 is depicted by the following diagram:

PRECEDING PAGE BLANK NOT FILMED

PAGE 71 INTENTIONALLY BLANK



In static analysis we are concerned only with the grid point set, $\{u_g\}$, and its subsets. The application of constraints and partitioning to the stiffness matrix involves, essentially, the elimination of $\{u_m\}$, $\{u_s\}$, $\{u_o\}$ and $\{u_r\}$ from $\{u_g\}$ to form a stiffness matrix referred to $\{u_\ell\}$.

The physical and computational significances of these operations are explained in Section 3.5. For the present it will only be emphasized that the concept of nested vector sets is extremely important in the theoretical development of NASTRAN. The reader may, in fact, find it useful at some point to memorize the relations, defined in Table 2, among the displacement sets.

Load vectors are distinguished by the same notation. Rectangular matrices are, whenever necessary to clarify the meaning of the symbol, distinguished by double subscripts referring to the vector sets associated with the rows and columns of the array. Superscripts have no tensorial character and are used to identify arrays of different type or origin that refer to the same sets such as in the equation,

$$[M_{dd}] = [M^1_{dd} + M^2_{dd}] \quad , \tag{2}$$

where $[M^1_{dd}]$ is the structural mass matrix and $[M^2_{dd}]$ is the direct input mass matrix.

Two types of operations occur repeatedly. These are the partitioning (or sort) operation, for example,

$$\{u_g\} = \left\{\begin{matrix} u_n \\ \hdashline u_m \end{matrix}\right\} \quad , \tag{3}$$

and the recombining (or merge) operation

$$\left\{\begin{matrix} u_n \\ \hdashline u_m \end{matrix}\right\} \rightarrow \{u_g\} \; . \tag{4}$$

In the sort operation indicated, the elements of $\{u_g\}$ are sorted into two lists. In the merge operation $\{u_n\}$ and $\{u_m\}$ are combined into a single list. In all sort and merge operations the resulting arrays are ordered according to the grid point sequence numbers of the displacement components.

In addition to the formal symbols used in matrix operations, many other symbols are required in the reduction of physical properties to matrix form. No special system is used for the latter class of symbols. An attempt has been made, however, to adhere to established engineering conventions.

Table 1. Root Symbols Used in Static Analysis in NASTRAN.

$\{u\}$	vector of displacement components
$\{P\}$	vector of applied load components
$\{q\}$	vector of forces of reaction
$\{Y\}$	vector of enforced displacements
$[K]$	stiffness matrix
$[M]$	mass matrix
$[B]$	damping matrix
$[R]$	matrix of constraint coefficients, as in $[R]\{u\} = 0$
$[G]$	transformation matrix, as in $\{u_m\} = [G_m]\{u_n\}$
$[D]$	rigid body transformation matrix
$[m]$	rigid body mass matrix
$[X]$	rigid body stiffness matrix
$[L]$	lower triangular factor of $[K]$
$[U]$	upper triangular factor of $[K]$

NOTATION SYSTEM

Table 2a. Mutually Independent Vector Sets.

u_m	coordinates eliminated as independent degrees of freedom by multi-point constraints and rigid elements
u_s	coordinates eliminated by single point constraints
u_o	coordinates omitted by structural matrix partitioning
u_r	coordinates to which determinate reactions are applied in static analysis
u_ℓ	the remaining structural coordinates used in static analysis (points left over)
u_e	extra degrees of freedom introduced in dynamic analysis to describe control systems, etc.

Table 2b. Combined Vector Sets.

$u_a = u_r + u_\ell$, the set used in real eigenvalue analysis

$u_d = u_a + u_e$, the set used in dynamic analysis by the direct method

$u_f = u_a + u_o$, unconstrained (free) structural coordinates

$u_n = u_f + u_s$, all structural coordinates not constrained by multi-point constraints or rigid elements

$u_g = u_n + u_m$, all structural (grid) points including scalar points

$u_p = u_g + u_e$, all physical coordinates

Note: (+) sign indicates the union of sets.

Table 2c. Modal Coordinate Sets.

ξ_o	rigid body (zero frequency) modal coordinates
ξ_f	finite frequency modal coordinates

$\xi_i = \xi_o + \xi_f$, the set of all modal coordinates.

$u_h = \xi_i + u_e$, the set used in dynamic analysis by the modal method.

Note: (+) sign indicates the union of sets.

Table 2d. Aerodynamic Coordinate Sets.

u_k aerodynamic box and body coordinates

u_{sA} permanently constrained aerodynamic coordinates

$u_A = u_k + u_{sA}$, all aerodynamic coordinates

$u_{ps} = u_p + u_{sA}$

$u_{pA} = u_p + u_A$, all physical and aerodynamic coordinates

Note: (+) sign indicates the union of sets.

3.4 PROBLEM FORMULATION

The explanation of any complex activity must be subdivided into phases or steps in order to be intelligible. In the case of a computer program for structural analysis it is convenient to divide the total effort into a Problem Formulation Phase and a Problem Solution Phase. The termination of the Problem Formulation Phase is arbitrarily chosen to occur at the point where the properties of the structure have been reduced to matrix form. (In the case of basic static analysis this occurs between blocks 4 and 5 in the flow diagram of Figure 1, Section 3.2.)

3.4.1 Structural Modeling

The beginning of the Problem Formulation Phase occurs in the mind of the analyst. He contemplates the problem, decides what he needs to know, and constructs a mathematical model whose solution, he hopes, will provide relevant answers to his questions. He will, naturally, require computational tools to solve his mathematical problem and, fortunately or unfortunately, the available tools have a strong influence on the analyst's choice of a mathematical problem. It would, after all, do no good to formulate a problem that could not be solved.

The range of choice in mathematical problem formulation provided by NASTRAN is, however rich in detail, limited to one basic approach, namely the use of finite element structural models. This means that the substitute mathematical problem refers to an idealized model with a finite number of degrees of freedom, a particular selection of topological objects (grid points and elements), and a limited range of structural behavior. The relevance of the behavior of the idealized structural model to the analyst's questions clearly depends on the particular choice of components for the model. This procedure, referred to as "structural modeling," is the most important step in the problem formulation phase, since the results of an analysis can be no better than the initial assumptions.

The User's Manual contains a chapter on structural modeling. Section 14 of the Theoretical Manual describes some advanced modeling techniques that utilize special features of NASTRAN. For the present, a small example will serve to indicate the general nature of the modeling process and some of the features of NASTRAN that relate to it.

Figure 1a shows a typical aircraft structure, a ring frame with a partial bulkhead acting as a floor support. Although poor results are obtained when such structures are analyzed without considering the shell to which the frame is attached, the analyst may have a special reason for doing so. The resulting idealized model of the frame can, in any case, serve as part of the model for the complete shell.

The idealized model selected by the analyst, Figure 1b, contains

13 grid points

4 Bar elements (B)

2 Rod elements (R)

2 Triangular Plate Elements (T)

3 Quadrilateral Plate Elements (Q)

Each grid point has six degrees of freedom (three translations and three rotations). The analyst has, however, elected to analyze the response of the frame to a pair of vertical loads so that it is unnecessary to consider out-of-plane motions of the frame. The out-of-plane motions are eliminated by applying single point constraints to three of the degrees of freedom (two rotations and one translation) at each gridpoint (This can be implemented with a single data card).

One of the necessary tasks in preparing input data is to specify the location of grid points. In NASTRAN grid point locations can be specified by rectangular, cylindrical or spherical coordinate systems (see Figure 2) and there may be an unlimited number of coordinate systems of each type in a given problem. All that is required is that they be related, directly or indirectly, to each other and to a "basic" coordinate system, which is rectangular. In the example of Figure 1, the analyst found it convenient to locate grid points on the ring frame (points 1 to 4) with a cylindrical coordinate system and to locate points on the floor bulkhead (points 5 to 13) with a rectangular coordinate system.

A separate task is the selection of coordinate systems to express the components of motion at grid points. In the example of Figure 1, the coordinate systems for motion have been selected to be identical to the coordinate systems for grid point location, although this is not required.

It will be noted in Figure 1b that the grid points for the ring frame are located on the outer edge of the frame rather than along its centerline. This will not result in poor accuracy if the provision for offsetting the neutral axis of Bar elements is exercised. Reinforcing Rod elements (R1 and R2), which have axial stiffness only, are placed between grid points 11, 12, and 13 to simulate the stiffener along the centerline.

The Plate elements (T's and Q's) are selected to contain membrane (plane stress) properties only, since out of plane bending is precluded by the nature of the loading. No restraint on inplane rotation (θ_z) is provided by the plate elements so that the θ_z component of motion must be eliminated by more single point constraints at gridpoints 6 to 13. A special problem occurs at grid point 5 because of the requirement to maintain compatibility of inplane rotation between the adjacent bar element (B_4) and the adjacent triangular plate (T_1). The problem is solved by means of a multipoint constraint between inplane rotation (θ_z) at grid point 5 and the vertical motions (u_y) at grid points 5 and 6. The equation of constraint is

$$\theta_{z5} = \frac{u_{y5} - u_{y6}}{x_5 - x_6} \qquad (1)$$

Additional single point constraints are required along the centerline of symmetry to constrain motions in the x direction (including the θ direction at gridpoint 1). A special type of single point constraint, known as a reaction, is used to constrain vertical motion at grid point 13. Constraints of this type are automatically removed when a static analysis is followed by a dynamic analysis. In addition, a special check calculation is provided (see Section 3.5.5) to determine whether the input impedance at reaction points is correct.

It will be noted that the grid points in Figure 1 have been numbered consecutively starting at the top. More than a sense of orderliness is involved since the sequencing of grid point numbers affects the bandwidth of the stiffness matrix and the resulting computer solution time (see Section 2.2). Grid point sequencing strategy is discussed in the User's Manual. The main idea is that the arithmetic differences between the sequence numbers of grid points that are physically adjacent should be minimized.

In order to facilitate grid point sequencing for the preservation of bandwidth, the user is permitted to specify grid point numbers in two different ways. The external identification numbers can be assigned to grid points in any manner the user desires. Element connection and load information prepared by the user refers to the external identification numbers. The internal sequence numbers are generated by the user in a paired list that relates external and internal numbers. Since the internal sequence numbers appear nowhere else in the input data, they may easily be changed, if desired, to reflect an improved banding strategy. Preparation of the paired list is optional and the sequence of the external identification numbers will be used if the paired list is not provided.

Another sequencing feature of NASTRAN is the ability to insert new grid point sequence numbers anywhere in an established list. This is done by the use of Dewey decimal notation, similar to that found in public libraries.

3.4.2 Input Data Analysis

We have now arrived at the point in problem formulation where the digital computer appears on the scene. The user assembles the information discussed above (plus a great many details that were not mentioned) and enters it on punched cards that are input to the computer. In problems that have many grid points arranged in regular patterns he may elect to write a small auxiliary program that will prepare and punch most of the input data cards (or their card images on magnetic tape). Such "supermarket" programs (so called because they can produce a shopping cartload of data cards) are a regular internal feature of some structural analysis programs but not of NASTRAN. They were not included because they become quite intricate, and hence, difficult to use, as they are given the generality that is needed for diverse applications. It is easier, on the average, to write a new supermarket program for each type of application. The user can, by means of the ALTER feature (see Section 1.2), incorporate such subroutines into NASTRAN.

When assembled the NASTRAN data deck consists of the following four parts:

1. Executive Control Deck
2. Substructure Control Deck (optional)
3. Case Control Deck
4. Bulk Data Deck

The Executive Control Deck identifies the job and the type of solution to be performed. It also declares the general conditions under which the job is to be executed, such as, maximum time allowed, type of system diagnostics desired, restart conditions and whether or not the job is to be checkpointed. If the job is to be executed with a rigid format, the number of the rigid format is declared along with any alterrations to the rigid format that may be desired. If Direct Matrix Abstruction is used, the complete DMAP sequence must appear in the Executive Control Deck.

The Substructure Control Deck is included only when Automated Multi-stage Substructuring is used. It defines the general attributes of the Automated Multi-stage Substructuring capability and establishes the control of the Substructure Operating File (SØF)

The Case Control Deck defines the subcase structure for the problem, makes selections from the Bulk Data Deck, and makes output requests for printing, punching, and plotting. The subcase structure for each of the rigid formats is described in the User's Manual. Loading conditions, boundary conditions, and other items are selected from the Bulk Data Deck in order to define the structural model for each subcase.

The Bulk Data Deck contains all of the details of the structural model. Much of this deck is associated with the definition of the grid points (grid cards) and the manner of connecting the grid points with elements (connection cards).

A number of important preliminary operations are performed on the data deck by the Input File Processor. It sorts the Bulk Data Deck, and stores it on the New Problem Tape. It checks the data cards for fatal errors. It creates the data blocks used by functional modules. If fatal errors are detected, suitable error messages are written and the execution is terminated.

3.4.3 Geometry Processor and Structure Plotter

The various parts of the Geometry Processor (see Figure 1 of Section 3.2) perform the following general tasks:

1. Generate all required coordinate system transformation matrices and determine the locations of all grid points in the basic coordinate system.
2. Replace external grid point numbers with their internal (sequential) indices.
3. Generate multipoint constraint equations and lists of single-point constraints.
4. Generate flags indicating the displacement components which are members of each displacement vector set (see Section 2.1.5).

Grid points may be defined in terms of the basic coordinate system or in terms of "local" coordinate systems (see Section 3.4.1). The Geometry Processor calculates the location and orientation of each local coordinate system relative to the basic system. This information is saved for later use by the various modules in making coordinate system transformations. The basic system is used for plotting (see Section 13).

As explained in Section 3.4.1, coordinate systems for expressing components of motion can be freely selected so that, for example, each grid point may have a unique displacement coordinate system associated with it. The collection of all displacement component directions in their own coordinate systems is known as the "global" coordinate system. All matrices are formed and all displacements are calculated in the global coordinate system.

The Structure Plotter is run after the second phase of the Geometry Processor. At this stage of the execution there is sufficient geometrical information in suitable form to prepare a plot of the undeformed structure.

3.4.4 Assembly of Structrual Matrices

The Element Matrix Generator (EMG) and the Element Matrix Assembler (EMA) generate the stiffness, mass, and damping matrices for the structural model. For efficiency in restart, particularly when changing from statics to dynamics problems, the structural matrices, $[K_{gg}]$, $[K^4_{gg}]$, $[M_{gg}]$ and $[B_{gg}]$, are assembled by four separate executions of EMA. EMG generates the various types of structural matrices on a selective basis. A third part of the matrix assembly matrix operation (SMA3) adds the contributions of the general elements (see Section 5.7) to the stiffness matrix.

The Element Matrix Generator refers to the appropriate "element" routines for calculation of the stiffness, mass and damping matrices for each element. The elements available for use are described in Section 5. The matrices for each element are initially generated in an element coordinate system that is characteristic for each element type. The element matrices are transformed to the global coordinate system prior to transfer to direct access secondary storage.

The Element Matrix Assembler assembles several columns of the structural matrices at one time. The number of columns assembled in one operation is limited by the space available in main storage. The required element matrices are transferred from secondary storage using the direct access read operation. The completed columns of the structural matrices are written on secondary storage by using the regular NASTRAN pack routines.

Prior to writing the completed matrices for each grid point on secondary storage devices, they are checked for singularities at the grid point level. Singularities remaining at this level, following a check of a list of the single-point constraints and the dependent coordinates of the multipoint constraint equations and/or rigid elements, are treated as warnings to the user or, on option, are automatically constrained. They are treated only as warnings because it cannot be determined at the grid point level whether or not the singularities are removed by other means, such as by general elements or by multipoint constraints and/or rigid elements in which these singularities are associated with independent coordinates.

PROBLEM FORMULATION

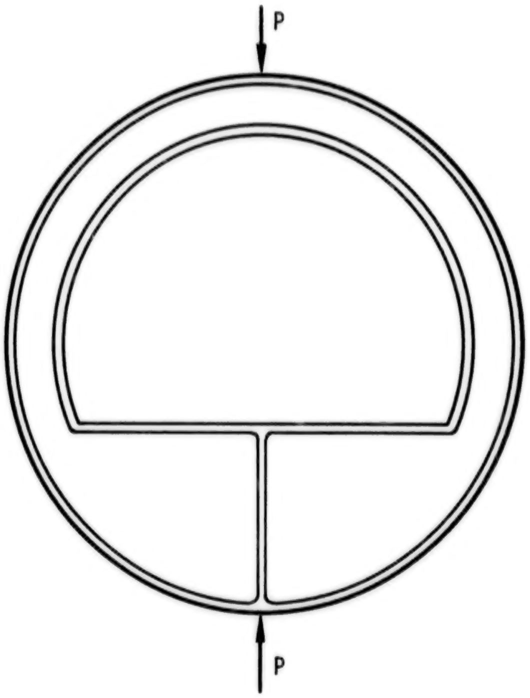
The user has the option* of requesting the program to automatically remove the strongest combination (weakest structure) singularities. However, this is not failsafe since the orientation of the actual grid point singularities may not be parallel to the displacement coordinate systems at the corresponding grid points and, in dynamic problems, the inertial forces due to the masses may be constrained incorrectly. Also, if the user has specified omitted points, rigid body support points, and/or multipoint constraint equations and rigid elements, these affected degrees of freedom may be overridden by requesting automatic removal of the strongest combination singularities.

Singularities are detected by examining the diagonal term for scalar grid points and the 3 x 3 matrices located along the diagonal of the stiffness matrix and associated with the rotational and translational degrees of freedom for geometric grid points. If the diagonal term for a scalar point is null, this fact is noted in the Grid Point Singularity Table (GPST). If either of the 3 x 3 matrices, associated with a geometric point, is singular, the diagonal terms and the 2 x 2 minors are examined to determine the order of singularity and the column or columns associated with the singularity. The order and locations of any singularities at geometric grid points are added to the GPST.

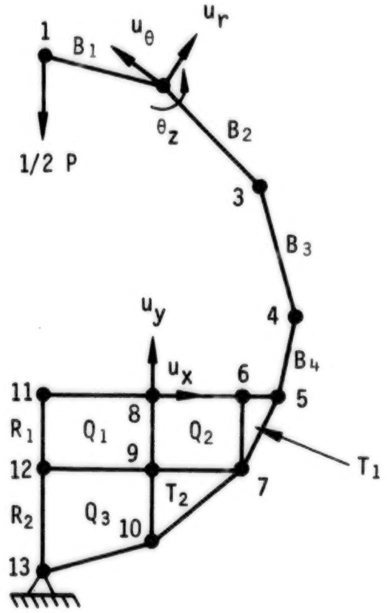
Although the matrices generated by the Structural Matrix Assembler are symmetric, complete columns are generated and retained for efficiency in succeeding matrix operations. This is necessary because all matrix operations are performed one column at a time (see Section 2) and in dynamics applications the matrices are not necessarily symmetric. Moreover, the availability of symmetric matrices by rows or by columns is advantageous in some of the matrix operations.

---

*At present, this option can be exercised only by means of a DMAP alter in the Executive Control Deck. See Section 5.10 of the User's Manual for details.



a. Ring frame with floor bulkhead.


1
B1
u_θ
u_r
θ_z
B2
1/2 P
3
B3
4
u_y
u_x
6
B4
11
5
8
R1
Q1
Q2
9
T1
12
7
T2
R2
Q3
10
13

b. Idealized structural model.

Figure 1. Example of structural modeling.

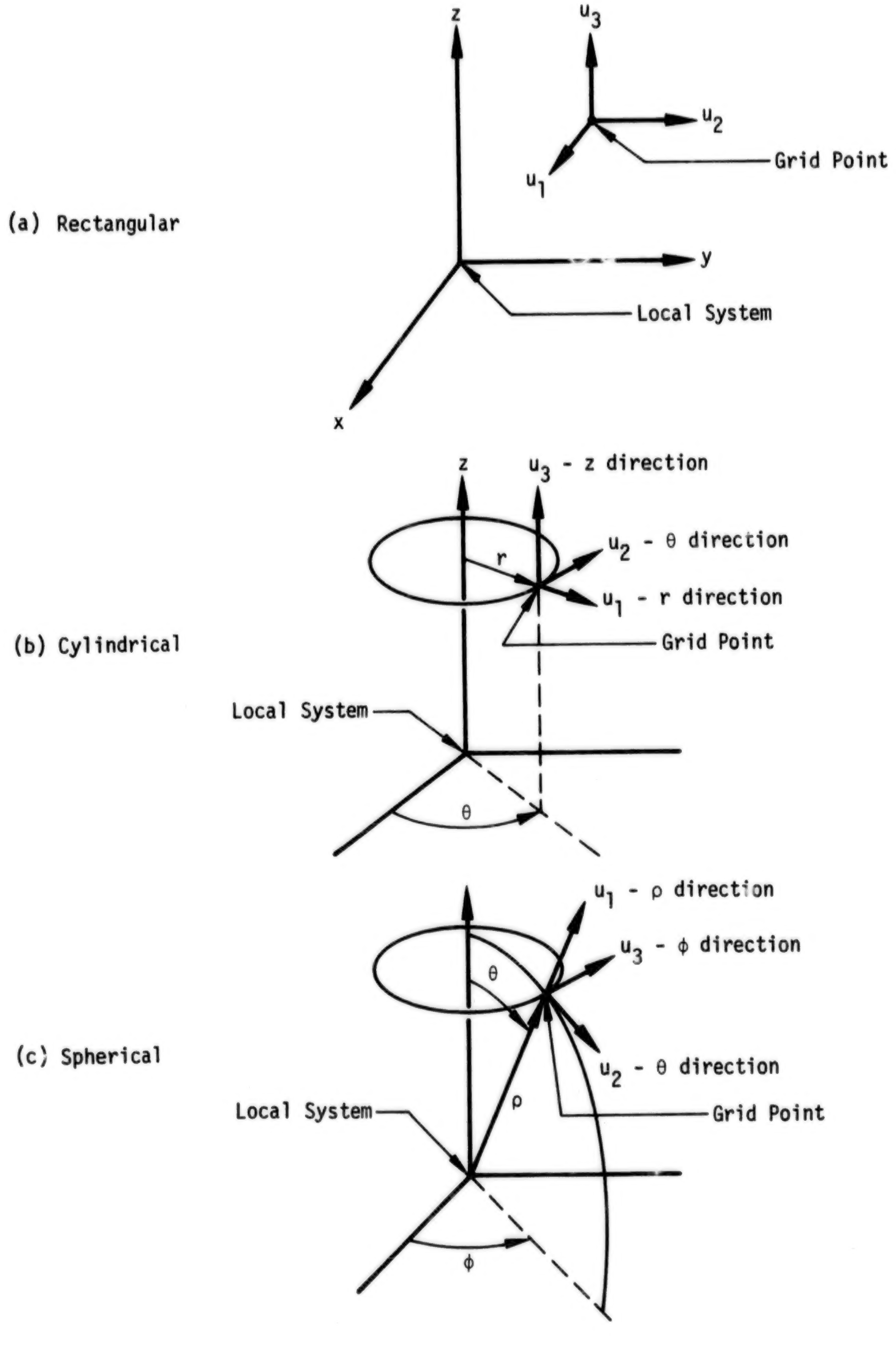

(a) Rectangular
z
u_3
u_2
u_1
Grid Point
y
Local System
x
(b) Cylindrical
z
u_3 - z direction
u_2 - θ direction
r
u_1 - r direction
Grid Point
Local System
θ
(c) Spherical
u_1 - ρ direction
u_3 - ϕ direction
θ
u_2 - θ direction
Local System
ρ
Grid Point
ϕ

Figure 2. Displacement coordinate systems.

3.5 CONSTRAINTS AND PARTITIONING

Structural matrices are initially assembled in terms of the set, u_g, of all structural grid points, which excludes only the set, u_e, of extra points used in dynamic analysis. This section will describe the subsequent reduction of the structural matrices to the set, u_ℓ, which is the set of coordinates that remain after all constraint and partitioning operations have been performed, and which is, therefore, the first set to be evaluated in static analysis.

The structural matrices whose assembly is discussed in the preceding section are:

$[K_{gg}]$	the structural stiffness matrix due to elastic structural elements
$[K^4_{gg}]$	the structural damping matrix of imaginary stiffness coefficients
$[B_{gg}]$	the viscous damping matrix due to damper elements
$[M_{gg}]$	the structural mass matrix

The reduction procedures will be explained in full for the $[K_{gg}]$ matrix. Procedures for the other matrices will be shown only when they differ from those for $[K_{gg}]$.

Repeated use will be made of the notation system described in Section 3.3, to which the reader's attention is directed.

3.5.1 Multipoint Constraints and Rigid Elements

Multipoint constraints and rigid elements are used to constrain one or more degrees of freedom to be equal to linear combinations of the values of other degrees of freedom. In the former case, the user must provide explicitly the coefficients of the constraint equations while, in the latter case, he needs to provide only the connection data; the program internally generates the required coefficients. The mathematical details of the generation of these coefficients are discussed in Section 3.5.6.

The constraint equations resulting from the use of multipoint constraints and rigid elements together can be expressed in the form

$$[R_g]\,\{u_g\} = 0, \qquad (1)$$

PAGE 87 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

ORIGINAL PAGE IS OF POOR QUALITY

where the coefficients are either supplied explicitly by the user (in the form of multipoint constraint equations) or are internally generated by the program (in the case of rigid elements; see Section 3.5.6). The user also specifies the degrees of freedom that are made dependent by the equations of multipoint constraint and by the rigid elements, so that the $\{u_g\}$ matrix may immediately be partitioned into two subsets,

$$\{u_g\} = \begin{Bmatrix} u_n \\ \hdashline u_m \end{Bmatrix} , \qquad (2)$$

where the set, u_m, is the set of dependent degrees of freedom. The matrix of constraint coefficients is similarly partitioned

$$[R_g] = [R_n \mid R_m] , \qquad (3)$$

so that Equation 1 becomes

$$[R_n]\{u_n\} + [R_m]\{u_m\} = 0 . \qquad (4)$$

$[R_m]$ is a nonsingular matrix. The constraint matrix can, therefore, be formed as

$$[G_m] = -[R_m]^{-1}[R_n], \qquad (5)$$

so that Equation 4 may be stated as

$$\{u_m\} = [G_m]\{u_n\} . \qquad (6)$$

Prior to the imposition of constraints, the structural problem may be written as

$$[K_{gg}]\{u_g\} = \{P_g\} , \qquad (7)$$

or, partitioning in terms of the coordinate sets, u_n and u_m

$$\begin{bmatrix} \bar{K}_{nn} & K_{nm} \\ K_{nm}^T & K_{mm} \end{bmatrix} \begin{Bmatrix} u_n \\ u_m \end{Bmatrix} = \begin{Bmatrix} \bar{P}_n \\ P_m \end{Bmatrix} . \qquad (8)$$

Bars over symbols are used to designate arrays that are replaced in the reduction process.

The addition of constraints to the structure requires that the forces of constraint be added to the equilibrium equations. It is shown in Section 5.4 that the forces of constraint are proportional to the corresponding coefficients in the constraint equations. Thus, writing the equilibrium and constraint equations together in partitioned form,

$$\left[\begin{array}{c|c|c} \bar{K}_{nn} & K_{nm} & G_m^T \\ \hline K_{nm}^T & K_{mm} & -I \\ \hline G_m & -I & 0 \end{array}\right] \left\{\begin{array}{c} u_n \\ \hline u_m \\ \hline q_m^m \end{array}\right\} = \left\{\begin{array}{c} \bar{P}_n \\ \hline P_m \\ \hline 0 \end{array}\right\} , \tag{9}$$

where $\{q_m^m\}$ is the vector of constraint forces on $\{u_m\}$. Straightforward elimination of u_m and q_m^m gives

$$[\bar{K}_{nn} + K_{nm}G_m + G_m^T K_{nm}^T + G_m^T K_{mm} G_m]\{u_n\} = \{\bar{P}_n\} + [G_m]^T\{P_m\} , \tag{10}$$

or

$$[K_{nn}]\{u_n\} = \{P_n\} , \tag{11}$$

where

$$[K_{nn}] = [\bar{K}_{nn} + K_{nm}G_m + G_m^T K_{nm}^T + G_m^T K_{mm} G_m] , \tag{12}$$

and

$$\{P_n\} = \{\bar{P}_n\} + [G_m]^T\{P_m\} . \tag{13}$$

The initial partition of K_{gg} and the operations indicated by Equations 5, 12 and 13 are performed by appropriate modules of the program. The constraint matrix, G_m, is used in structural matrix reduction (Equation 12), load vector reduction (Equation 13) and data recovery (Equation 6). It is saved for these purposes in an auxiliary storage file.

The other structural matrices, $[K_{gg}^4]$, $[B_{gg}]$ and $[M_{gg}]$, are transformed by formulas that are identical in form to Equation 12.

PRECEDING PAGE BLANK NOT FILMED

PAGE 3.5-1B INTENTIONALLY BLANK

The forces of constraint due to multipoint constraints and rigid elements are not available as part of the standard calculations in the program, but may be obtained in statics and real eigenvalue problems by ALTERing in the module EQMCK and requesting these forces in the Case Control Deck.

The vector of constraint forces $\{q_m^m\}$ on $\{u_m\}$ is obtained from the middle row of Equation 9 as

$$\{q_m^m\} = -\{P_m\} + [K_{nm}]^T \{u_n\} + [K_{mm}]\{u_m\} \quad . \tag{13a}$$

Let q_n^m be the vector of constraint forces on $\{u_n\}$. Then, from the first row of Equation 9, it is clear that

$$\{q_n^m\} = -[G_m]^T \{q_m^m\} \quad . \tag{13b}$$

The resultant forces of constraint due to multipoint constraints and rigid elements are therefore given by

$$\{q_g^m\} = \left\{\begin{array}{c} q_n^m \\ \hline q_m^m \end{array}\right\} \quad . \tag{13c}$$

3.5.2 Single Point Constraints

Single point constraints are applied to the set, u_s, in the form

$$\{u_s\} = \{Y_s\} \ , \tag{14}$$

where $\{Y_s\}$ is a vector of enforced deformations, any or all of whose elements may be zero. The set, u_n, is partitioned into u_s and u_f(the free or unconstrained set)

$$\{u_n\} = \left\{\begin{array}{c} u_f \\ \hline u_s \end{array}\right\} . \tag{15}$$

The stiffness matrix, K_{nn}, is similarly partitioned

$$[K_{nn}] = \left[\begin{array}{c|c} K_{ff} & K_{fs} \\ \hline K_{fs}^T & K_{ss} \end{array}\right] \quad . \tag{16}$$

The complete structural equations including the single point forces of constraint, q_s, may be written in partitioned matrix form as

ORIGINAL PAGE IS OF POOR QUALITY

$$\left[\begin{array}{c|c|c} K_{ff} & K_{fs} & 0 \\ \hline K_{fs}^T & K_{ss} & -I \\ \hline 0 & I & 0 \end{array}\right]\left\{\begin{array}{c} u_f \\ \hline u_s \\ \hline q_s \end{array}\right\} = \left\{\begin{array}{c} \bar{P}_f \\ \hline P_s \\ \hline Y_s \end{array}\right\} . \tag{17}$$

Straightforward elimination gives

$$[K_{ff}]\{u_f\} = \{\bar{P}_f\} - [K_{fs}]\{Y_s\} = \{P_f\}. \tag{18}$$

The forces of constraint are recovered by means of the middle row of Equation 17, i.e.,

$$\{q_s\} = -\{P_s\} + [K_{fs}^T]\{u_f\} + [K_{ss}]\{u_s\} . \tag{19}$$

Thus all three of the distinct partitions of K_{nn} (i.e., K_{ff}, K_{fs} and K_{ss}) are needed in subsequent calculations, and are placed in auxiliary storage. For the other structural matrices (K_{nn}^4, B_{nn}, and M_{nn}) only the (ff) partitions are saved. The assumption is made, implicitly, that the effects of the other structural matrices on the single point forces of constraint may be ignored.

3.5.3 Partitioning

At user option the set of free coordinates, u_f, may be partitioned into two sets, u_o and u_a, such that the u_o set is eliminated first. Thus

$$\{u_f\} = \left\{\begin{array}{c} u_a \\ \hline u_o \end{array}\right\} . \tag{20}$$

The equilibrium equations after the elimination of constraints (Equation 18) may be written in partitioned form as

$$\left[\begin{array}{c|c} \bar{K}_{aa} & K_{ao} \\ \hline K_{ao}^T & K_{oo} \end{array}\right]\left\{\begin{array}{c} u_a \\ \hline u_o \end{array}\right\} = \left\{\begin{array}{c} \bar{P}_a \\ \hline P_o \end{array}\right\} . \tag{21}$$

Rearrange the bottom half of Equation 21:

$$[K_{oo}]\{u_o\} = \{P_o\} - [K_{ao}]^T\{u_a\} , \tag{22}$$

and solve for $\{u_o\}$:

$$\{u_o\} = [K_{oo}]^{-1}\{P_o\} - [K_{oo}]^{-1}[K_{ao}]^T\{u_a\} . \tag{23}$$

ORIGINAL PAGE IS
OF POOR QUALITY

(Note that in practice stiffness matrices are never inverted due to excessive computer running time. The practical alternative will be explained presently.)

Substitute for u_o into the top half of Equation 21:

$$[\bar{K}_{aa} - K_{ao}K_{oo}^{-1}K_{ao}^{T}]\{u_a\} = \{\bar{P}_a\} - [K_{ao}][K_{oo}]^{-1}\{P_o\} . \tag{24}$$

It is convenient to define the matrix

$$[G_o] = -[K_{oo}]^{-1}[K_{ao}]^{T} , \tag{25}$$

so that Equation 24 becomes

$$[\bar{K}_{aa} + K_{ao}G_o]\{u_a\} = \{\bar{P}_a\} + [G_o]^{T}\{P_o\} , \tag{26}$$

where advantage is taken of the symmetry of $[K_{oo}]$.

Following the practice of condensation established in preceding subsections,

$$[K_{aa}]\{u_a\} = \{P_a\} , \tag{27}$$

where

$$[K_{aa}] = [\bar{K}_{aa}] + [K_{ao}][G_o] , \tag{28}$$

$$\{P_a\} = \{\bar{P}_a\} + [G_o]^{T}\{P_o\} . \tag{29}$$

The $[G_o]$ matrix defined in Equation 25 is obtained practically from the solution of

$$[K_{oo}][G_o] = -[K_{ao}]^{T} , \tag{30}$$

where $[K_{ao}]^T$ is treated as a set of load vectors. Each such vector produces a column of $[G_o]$. The $[K_{oo}]$ matrix is first decomposed into lower and upper triangular factors, using a subroutine based on the techniques described in Section 2.2. The additional steps required in solving the matrix equation $[A]\{x\} = \{b\}$ are described in Section 2.3.

Once $\{u_a\}$ is obtained the set of omitted coordinates, $\{u_o\}$, is obtained as follows. Define the set $\{u_o^o\}$ as the solution of

$$[K_{oo}]\{u_o^o\} = \{P_o\} . \tag{31}$$

Note that the triangular factors of $[K_{oo}]$ obtained in connection with Equation 30 are saved for use in connection with Equation 31 which cannot be solved until the load vector $\{P_o\}$ is formed. Then, using Equations 25 and 31 in Equation 22,

$$\{u_o\} = \{u_o^o\} + [G_o]\{u_a\} \quad . \tag{32}$$

Partitioning, which is an optional feature of the program, has a number of important uses. The first is as an aid to improved efficiency in the solution of ordinary static problems where it functions as an alternative to the Active Column technique (see Section 2.2) in reducing matrix bandwidth. In this application the user puts into the set u_a those degrees of freedom that are excessively coupled to the remainder.

In a related application, members of the set u_a are placed along lines or in planes of the structure such that the remaining u_o grid points in different regions are uncoupled from each other as shown in the wing structure of Figure 1. The grid points are sequenced so that all grid points in region (1) precede those in region (2), etc. As a result the decomposition of $[K_{oo}]$ is faster because the bandwidth is smaller (reduced to approximately 1/3 in the example). The u_a set is small compared to u_o so that its solution is not particularly time consuming. Even here proper grid point sequencing can introduce banding into the $[K_{aa}]$ matrix.

Matrix partitioning also improves efficiency when solving a number of similar cases with stiffness changes in local regions of the structure. The u_a and u_o sets are selected so that the structural elements that will be changed are connected only to grid points in the u_a set. The $[K_{oo}]$ matrix is then unaffected by the structural changes and only the smaller $[K_{aa}]$ matrix need be decomposed for each case. An application of partitioning that is important for dynamics is the Guyan Reduction, described in the next subsection.

3.5.4 The Guyan Reduction

The Guyan Reduction (Reference 1) is a means for reducing the number of degrees of freedom used in dynamic analysis with minimum loss of accuracy. Its basis is that many fewer grid points are needed to describe the inertia of a structure than are needed to describe its elasticity with comparable accuracy. If inertia properties are rationally redistributed to a smaller set of grid points, the remaining grid points can be assigned to the u_o set described in the preceding subsection and eliminated, leaving only the smaller u_a set for dynamic analysis.

In the Guyan Reduction, the means by which inertia (and damping) are redistributed is to consider the $[G_o]$ matrix of the preceding subsection as a set of rigid constraints, such that

$$\{u_o\} = [G_o]\{u_a\} \quad . \tag{33}$$

The $[G_o]$ matrix now has the same implications for the u_o coordinates that the multipoint constraint matrix, $[G_m]$, has for the u_m coordinates (see Equation 6). The reduced structural mass matrix is, by analogy with Equation 12,

$$[M_{aa}] = [\bar{M}_{aa} + M_{ao}G_o + G_o^T M_{ao}^T + G_o^T M_{oo} G_o] \quad . \tag{34}$$

The reduced damping matrices, $[K_{aa}^4]$ and $[B_{aa}]$, are formed in the same manner. The structural stiffness matrix, $[K_{aa}]$, is given by Equation 28. The reduced dynamic load vector is, by analogy with Equation 13,

$$\{P_a\} = \{\bar{P}_a + G_o^T P_o\} \quad . \tag{35}$$

The approximation made in the Guyan Reduction is that the term $\{u_o^o\}$ in Equation 32 is neglected; i.e. that the deformations of the u_o set relative to the u_a set due to inertia and other loads applied to the u_o set are neglected. The error in the approximation is small provided that the u_a set is judiciously chosen. The selection should be based, in part, on an estimate of the relative deformations, $\{u_o^o\}$. Thus the members of u_a should be uniformly dispersed throughout the structure and should include all large mass items. The basic assumption made in the Guyan Reduction is identical to that made in forming consistent mass matrices for individual elements, see Section 5.5.

3.5.5 Special Provisions for Free Bodies

A free body is defined as a structure that is capable of motion without internal stress. The stiffness matrix for a free body is singular with the defect equal to the number of stress-free (or free body) modes. A solid three-dimensional body has six or fewer free body modes. Linkages and mechanisms can have a greater number. No restriction is placed in the program on the number of stress-free modes in order to permit the analysis of mechanisms.

The presence of free body modes alters the details of many of the calculations in structural analysis. In static analysis by the displacement method, for example, the free body modes must be restrained in order to remove the singularity of the stiffness matrix. We are concerned, in this section, with some of the special provisions of the program for the treatment of free bodies including the specification of determinate reactions for use in static analysis, the evaluation of the inertia properties of free body modes for use in dynamic analysis, and a special diagnostic procedure for the detection of inconsistent constraints. Other special provisions are the calculation of inertia relief loads, treated in Section 3.6.3, and the procedures employed in the mode acceleration method of dynamic data recovery, treated in Section 9.4.

If a problem concerning a free body includes both static and dynamic solution cases, a subset, u_r, of the displacement vector, u_a, must be constrained during static analysis. The subset, u_r, is specified by the user such that the members of the set are just sufficient to eliminate the stress-free motions without introducing redundant constraints. The complete static equilibrium equations are

$$[K_{aa}]\{u_a\} = \{P_a\} , \tag{36}$$

or, partitioning u_a into u_r and u_ℓ,

$$\left[\begin{array}{c|c} K_{\ell\ell} & K_{\ell r} \\ \hline K_{\ell r}^T & K_{rr} \end{array}\right]\left\{\begin{array}{c} u_\ell \\ \hline u_r \end{array}\right\} = \left\{\begin{array}{c} P_\ell \\ \hline P_r \end{array}\right\} . \tag{37}$$

In static analysis the u_r set is rigidly constrained to zero motion so that the final problem solved in static analysis is

$$[K_{\ell\ell}]\{u_\ell\} = \{P_\ell\} . \tag{38}$$

The forces of reaction, $\{q_r\}$, which are of interest in their own right and which are also needed in the solution of inertia relief problems, are evaluated from the equation

$$\{q_r\} = -\{P_r\} + [K_{\ell r}]^T\{u_\ell\} , \tag{39}$$

or, substituting for $\{u_\ell\}$ from the solution to Equation 38,

$$\{q_r\} = -\{P_r\} + [K_{\ell r}]^T[K_{\ell\ell}]^{-1}\{P_\ell\} . \tag{40}$$

It is convenient to define the matrix

$$[D] = -[K_{\ell\ell}]^{-1}[K_{\ell r}] , \tag{41}$$

so that, taking advantage of the symmetry of $[K_{\ell\ell}]$,

$$\{q_r\} = -\{P_r\} - [D]^T\{P_\ell\} . \tag{42}$$

The [D] matrix is also used in the evaluation of the free body inertia properties of the structure. It is obtained practically by solution of the matrix equation

$$[K_{\ell\ell}][D] = [K_{\ell r}] . \tag{43}$$

$[K_{\ell\ell}]$ is decomposed into triangular factors, $[L_{\ell\ell}]$ and $[U_{\ell\ell}]$, which are saved and used in the solution of Equation 38 after the load vector $\{P_\ell\}$ has been evaluated.

It may be seen from Equation 37 that, in the absence of forces on the u_ℓ coordinates,

$$\{u_\ell\} = -[K_{\ell\ell}]^{-1}[K_{\ell r}]\{u_r\} = [D]\{u_r\} . \tag{44}$$

Thus the [D] matrix expresses the rigid body motions of the structure in response to displacements imposed at the reaction points.

The mass matrix, partitioned according to the u_ℓ and u_r sets, is

$$[M_{aa}] = \left[\begin{array}{c|c} M_{\ell\ell} & M_{\ell r} \\ \hline M_{\ell r}^T & M_{rr} \end{array}\right] . \tag{45}$$

If Equation 44 is taken as an equation of constraint for free body motion, the reduced mass matrix referred to the u_r coordinates is, by analogy with Equation 34,

$$[m_r] = [M_{rr} + M_{\ell r}^T D + D^T M_{\ell r} + D^T M_{\ell\ell} D] . \tag{46}$$

The free body mass matrix, $[m_r]$, and the rigid body transformation matrix, [D], complete the specification of the free body inertia properties that are used in dynamic analysis.

It is desirable to have a check on the compatibility of the single point and multipoint constraints previously placed on the structure (including the constraints imposed by the rigid elements) with the constraints placed on the reaction points, u_r. Such a check is obtained by noting that, if the u_ℓ set is eliminated from Equation 37, the reduced stiffness matrix referred to the u_r set should be completely null. The reduced stiffness matrix is

$$[X] = [K_{rr} - K_{\ell r}{}^{T} K_{\ell\ell}{}^{-1} K_{\ell r}]$$

$$= [K_{rr} + K_{\ell r}{}^{T} D] \quad . \tag{47}$$

The [X] matrix is computed by the program and its largest term is given to the user so that he may take appropriate action. No automatic test is built into the program. The [X] matrix may be nonzero for any of the following reasons:

1. Round-off error accumulation
2. $\{u_r\}$ is overdetermined (redundant supports)
3. $\{u_r\}$ is underdetermined ($K_{\ell\ell}$ is singular)
4. The multipoint constraints and/or rigid element connectivities are incompatible.
5. There are too many single point constraints.

3.5.6 Rigid Elements

3.5.6.1 Introduction

Rigid elements provide a convenient means of specifying very stiff connections. The user does not provide the required coefficients of the constraint equations directly. The program internally generates them from the connection data.

Four rigid elements are presently available. One of them is a rigid pin-ended rod element (RIGDR) and the other three are rigid body elements (RIGD1, RIGD2, and RIGD3). The use of these elements is discussed in Section 1.4.2.2 of the User's Manual.

3.5.6.2 The RIGDR Element

The RIGDR element (specified by the CRIGDR bulk data card) represents a pin-ended connection between two grid points that is rigid in extension-compression.

Let A and B be two grid points connected by a RIGDR element and let u_{A_1}, u_{A_2}, u_{A_3} and u_{B_1}, u_{B_2}, u_{B_3} represent the translational components of motion (in the basic coordinate system) at these points respectively. Let ℓ_1, ℓ_2, and ℓ_3 be the direction cosines (with respect to the basic coordinate system) of the line joining A to B. Then, since the distance between the points A and B remains unchanged, the following condition is satisfied for small displacements:

$$(u_{A_1} - u_{B_1})\ell_1 + (u_{A_2} - u_{B_2})\ell_2 + (u_{A_3} - u_{B_3})\ell_3 = 0 \quad , \tag{48}$$

or, in matrix form,

$$[\ell_1\ \ell_2\ \ell_3] \begin{Bmatrix} u_{A_1} \\ u_{A_2} \\ u_{A_3} \end{Bmatrix} = [\ell_1\ \ell_2\ \ell_3] \begin{Bmatrix} u_{B_1} \\ u_{B_2} \\ u_{B_3} \end{Bmatrix} \quad . \tag{49}$$

Let u'_{A_1}, u'_{A_2}, u'_{A_3} and u'_{B_1}, u'_{B_2}, u'_{B_3} be the translational components of motion at A and B in in their respective local displacement coordinate systems. These are related to the motion in the basic coordinate system by the equations

$$\begin{Bmatrix} u_{A_1} \\ u_{A_2} \\ u_{A_3} \end{Bmatrix} = [T_A] \begin{Bmatrix} u'_{A_1} \\ u'_{A_2} \\ u'_{A_3} \end{Bmatrix} \quad , \tag{50}$$

and

$$\begin{Bmatrix} u_{B_1} \\ u_{B_2} \\ u_{B_3} \end{Bmatrix} = [T_B] \begin{Bmatrix} u'_{B_1} \\ u'_{B_2} \\ u'_{B_3} \end{Bmatrix} \quad , \tag{51}$$

where $[T_A]$ and $[T_B]$ are (3 x 3) transformation matrices from the respective local displacement coordinate systems to the basic coordinate system.

Substitution of Equations 50 and 51 in Equation 49 gives

$$[\ell_1\ \ell_2\ \ell_3][T_A] \begin{Bmatrix} u'_{A_1} \\ u'_{A_2} \\ u'_{A_3} \end{Bmatrix} = [\ell_1\ \ell_2\ \ell_3][T_B] \begin{Bmatrix} u'_{B_1} \\ u'_{B_2} \\ u'_{B_3} \end{Bmatrix} \quad . \tag{52}$$

The above equation can be rewritten as

$$[\ell_{A_1}\ \ell_{A_2}\ \ell_{A_3}] \begin{Bmatrix} u'_{A_1} \\ u'_{A_2} \\ u'_{A_3} \end{Bmatrix} = [\ell_{B_1}\ \ell_{B_2}\ \ell_{B_3}] \begin{Bmatrix} u'_{B_1} \\ u'_{B_2} \\ u'_{B_3} \end{Bmatrix} \quad , \tag{53}$$

where ℓ_{A_1}, ℓ_{A_2}, ℓ_{A_3} and ℓ_{B_1}, ℓ_{B_2}, ℓ_{B_3} represent the modified direction cosines and are given by

$$[\ell_{A_1}\ \ell_{A_2}\ \ell_{A_3}] = [\ell_1\ \ell_2\ \ell_3][T_A] \quad , \tag{54}$$

and

$$[\ell_{B_1}\ \ell_{B_2}\ \ell_{B_3}] = [\ell_1\ \ell_2\ \ell_3][T_B] \quad . \tag{55}$$

Equation 53 is the single equation of constraint that represents a rigid pin-ended rod element connection between the grid points A and B. Note that only the three translational components of motion at each of the two points are involved in this equation. The rotations at the points are not involved. Any one of the six translational components may be specified as the dependent degree of freedom in a RIGDR element. The other five translational components are considered as reference degrees of freedom. This is summarized in Table 1.

If Equation 53 is to be valid, it is necessary that the grid points A and B be non-coincident. Otherwise, the direction cosines ℓ_1, ℓ_2, and ℓ_3 will be undefined. The program checks for this condition.

Equation 53 will also not be meaningful if the direction of motion defined by the dependent translational degree of freedom is perpendicular (or nearly perpendicular) to the rod element because, in that case, the corresponding modified direction cosine will be zero (or nearly zero). The program checks for this condition also.

3.5.6.3 The RIGD1 and RIGD2 Elements

The RIGD1 and RIGD2 elements (specified by the CRIGD1 and CRIGD2 bulk data cards) are similar in that they both involve a single reference grid point and one or more dependent grid points. The RIGD1 element is the simpler and defines a rigid element connection in which all six degrees of freedom of each of the dependent grid points are coupled to all six degrees of freedom of the reference grid point. The RIGD2 element is more general and defines a rigid element connection in which selected degrees of freedom of the dependent grid points are coupled to all six degrees of freedom of the reference grid point.

Consider a dependent grid point A that is rigidly coupled by means of a RIGD1 or RIGD2 element to a reference grid point B. For small displacements, the motion $\{u_A\}$ at the point A is related to the motion $\{u_B\}$ at the point B by the equation

$$\begin{Bmatrix} u_{A_1} \\ u_{A_2} \\ u_{A_3} \\ u_{A_4} \\ u_{A_5} \\ u_{A_6} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & (z_B-z_A) & -(y_B-y_A) \\ 0 & 1 & 0 & -(z_B-z_A) & 0 & (x_B-x_A) \\ 0 & 0 & 1 & (y_B-y_A) & -(x_B-x_A) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{B_1} \\ u_{B_2} \\ u_{B_3} \\ u_{B_4} \\ u_{B_5} \\ u_{B_6} \end{Bmatrix} , \qquad (56)$$

where the motions are in the basic coordinate system and x_A, y_A, z_A and x_B, y_B, z_B are the basic coordinates of the points A and B respectively.

Using relations similar to Equations 50 and 51, Equation 56 can be expressed in terms of the motion in the local displacement coordinate systems of A and B by

$$\begin{Bmatrix} u'_{A_1} \\ u'_{A_2} \\ u'_{A_3} \\ u'_{A_4} \\ u'_{A_5} \\ u'_{A_6} \end{Bmatrix} = \left[\begin{array}{c|c} T_A^{\,T} & 0 \\ \hline 0 & T_A^{\,T} \end{array}\right] \left[\begin{array}{c|c} I & \begin{matrix} 0 & \bar{z} & -\bar{y} \\ -\bar{z} & 0 & \bar{x} \\ \bar{y} & -\bar{x} & 0 \end{matrix} \\ \hline 0 & I \end{array}\right] \left[\begin{array}{c|c} T_B & 0 \\ \hline 0 & T_B \end{array}\right] \begin{Bmatrix} u'_{B_1} \\ u'_{B_2} \\ u'_{B_3} \\ u'_{B_4} \\ u'_{B_5} \\ u'_{B_6} \end{Bmatrix} , \qquad (57)$$

where T_A and T_B are (3 x 3) transformation matrices from the local displacement coordinate systems to the basic coordinate system and $\bar{x} = x_B-x_A$, $\bar{y} = y_B-y_A$ and $\bar{z} = z_B-z_A$. I is a (3 x 3) unit matrix.

Equation 57 can be written in compact form as

$$\{u'_A\} = [G]_{AB}\{u'_B\} \quad , \qquad (58)$$

where $[G]_{AB}$ is a (6 x 6) matrix. Each row of this $[G]_{AB}$ matrix corresponds to a dependent degree of freedom of grid point A, and each column corresponds to a reference degree of freedom of grid point B. Each element of this matrix represents a coefficient that corresponds to the coupling of a particular dependent degree of freedom of grid point A with a particular reference degree of freedom of grid point B.

Equation 58 defines a set of six linear equations of constraint that mathematically represent the rigid coupling of dependent grid point A to reference grid point B. In the case of a RIGD1 element, six equations of constraint are generated for each of the specified dependent grid points. In the case of a RIGD2 element, the equations generated correspond to those rows of $[G]_{AB}$ that represent the specified dependent degrees of freedom of grid point A.

Let m be the total number of dependent degrees of freedom specified on a RIGD1 or RIGD2 element. Then, the combination of Equations 58 for all of the dependent grid points yields m linear equations of constraint represented in matrix form by

$$\{u'\} = [G]_B\{u'_B\} \quad , \tag{59}$$

where $\{u'\}$ is an $(m \times 1)$ vector of dependent degrees of freedom (in global coordinate system) and $[G]_B$ is an $(m \times 6)$ matrix that represents the rigid coupling of the m dependent degrees of freedom to the six degrees of freedom of reference grid point B. Note that, in the case of a RIGD1 element, m is equal to six times the number of dependent grid points specified for the element. The above results are summarized in Table 1.

3.5.6.4 The RIGD3 Element

The RIGD3 element (specified by the CRIGD3 bulk data card) is the most general rigid element and defines a rigid connection in which selected degrees of freedom of the dependent grid points are coupled to six selected reference degrees of freedom. The six reference degrees of freedom can be selected at one or more (up to six) reference grid points, but they should together be capable of fully describing rigid body motion. In other words, the six reference degrees of freedom should be so selected that they together represent six independent components of motion. The program checks for this condition since otherwise it leads to the inversion of a singular matrix.

Let B be one of the (up to six) reference grid points in a RIGD3 element and let m be the total number of dependent degrees of freedom specified on the element. Then, for small displacements, just as in the case of a RIGD1 or RIGD2 element, the m equations of constraint can be expressed in terms of the motion of grid point B by the matrix equation

$$\{u'\} = [G]_B\{u'_B\} \quad , \tag{60}$$

which is a re-statement of Equation 59. Note, however, in this case that the six degrees of freedom of grid point B will not, in general, all be the required six reference degrees of freedom.

Hence, Equation 60 does not give the required constraint equations.

Let u'_{R_1}, u'_{R_2}, u'_{R_3}, u'_{R_4}, u'_{R_5}, and u'_{R_6} be the six specified reference degrees of freedom (at least some of which will be the degrees of freedom of grid point B) in the global coordinate system. Then, these six degrees of freedom are related to the motion of grid point B by the matrix equation

$$\{u'_R\} = [G]_{RB}\{u'_B\} \quad , \tag{61}$$

which is similar to Equation 60 and where $[G]_{RB}$ is a (6 x 6) matrix.

Equation 61 can be re-written as

$$\{u'_B\} = [G]^{-1}_{RB}\{u'_R\} \quad . \tag{62}$$

Note that $[G]^{-1}_{RB}$ will not exist if the six specified reference degrees of freedom do not together define six independent components of motion. The program checks for this condition.

Substitution of Equation 62 in Equation 60 yields

$$\{u'\} = [G]_B[G]^{-1}_{RB}\{u'_R\} \quad . \tag{63}$$

The above matrix equation gives the required equations of constraint for a RIGD3 element. This is summarized in Table 1.

3.5.6.5 Resultant Constraint Equations

The constraint equations for the rigid elements are generated in subroutine CRIGGP in module GP4. This routine computes the required constraint equations for all rigid elements in a model by means of Equation 53 (for all RIGDR elements), Equation 59 (for all RIGD1 and RIGD2 elements), and Equation 63 (for all RIGD3 elements). Module GP4 then combines these constraint equations for all rigid elements with the multipoint constraint equations supplied by the user to obtain the resultant constraint equations (Equation 1 in Section 3.5.1) for the model as a whole. Once the rigid elements and the multipoint constraint data are processed and the resultant constraint equations are obtained by module GP4, no distinction is subsequently made between those constraint equations that are due to rigid elements and those that are due to multipoint constraint data.

STATIC ANALYSIS BY THE DISPLACEMENT METHOD

REFERENCE

1. Guyan, R. J., "Reduction of Stiffness and Mass Matrices," AIAA Journal, Vol. 3, No. 2, February 1965, p. 380.

Table 1. Summary of Constraint Equations Generated for Rigid Elements

Name of element	Number of constraint equations generated (per element)	Number of ref. grid points (per element)	Number of ref. degrees of freedom involved (per element)	Constraint equations obtained from
RIGDR	1	1	5	Equation 53
RIGD1	6 x (Number of dependent grid points)	1	6	Equation 59
RIGD2	Number of dependent degrees of freedom	1	6	Equation 59
RIGD3	Number of dependent degrees of freedom	1 to 6	6	Equation 63

C-2

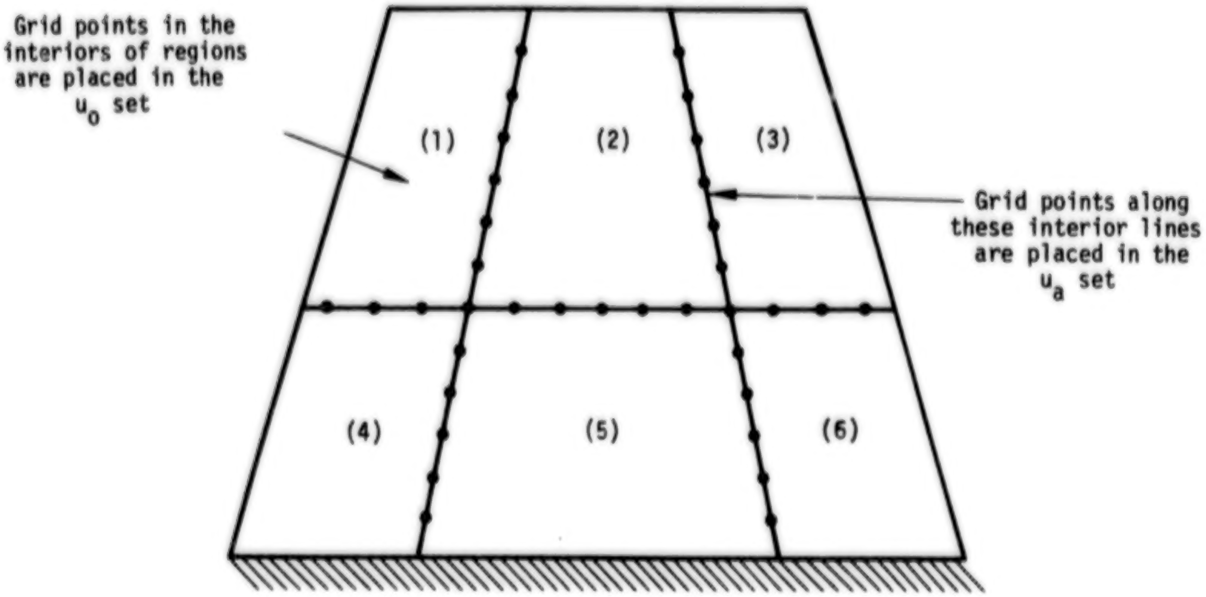

Grid points in the interiors of regions are placed in the u_o set
(1)
(2)
(3)
Grid points along these interior lines are placed in the u_a set
(4)
(5)
(6)

Figure 1. Use of partitioning to decouple regions of the structure.

3.6 STATIC LOADS

3.6.1 Generation of Loads

In NASTRAN, static loads are applied to geometric and scalar grid points in a variety of ways, including

1. Loads applied directly to grid points.
2. Pressure on surfaces.
3. Gravity loads, (internally generated).
4. Centrifugal forces due to steady rotation.
5. Equivalent loads resulting from thermal expansion.
6. Equivalent loads resulting from enforced deformations of structural elements.
7. Equivalent loads resulting from enforced displacements of grid points.

A force or a moment applied directly to a geometric grid point may be specified in terms of components along the axes of any coordinate system that has been defined. Alternatively, the direction of a force or a moment may be specified by a vector connecting a pair of specified grid points or as the cross-product of two such vectors. A load on a scalar point is specified by a single number since only one component of motion exists at a scalar point.

Pressure loads may be applied to triangular and quadrilateral plates and to axisymmetric shell elements. The positive direction of loading on a triangle is determined by the order of the corner grid points, using the right hand rule. The magnitude and direction of the load is automatically computed from the value of the pressure and the coordinates of the grid points. The load is divided equally to the three grid points.

The direction of pressure load on a quadrilateral plate is determined by the order of its corner grid points which need not lie in a plane. The grid point loads are calculated by dividing the quadrilateral into triangles in each of the two possible ways and applying one-half of the pressure to each of the four resulting triangles. Severely warped quadrilaterals should be subdivided into triangles by the user in order to provide better definition of the surface.

The user specifies a gravity load by providing the components of the gravity vector in any defined coordinate system. The gravitational acceleration of a translational component of motion,

a_i, at a geometric grid point is

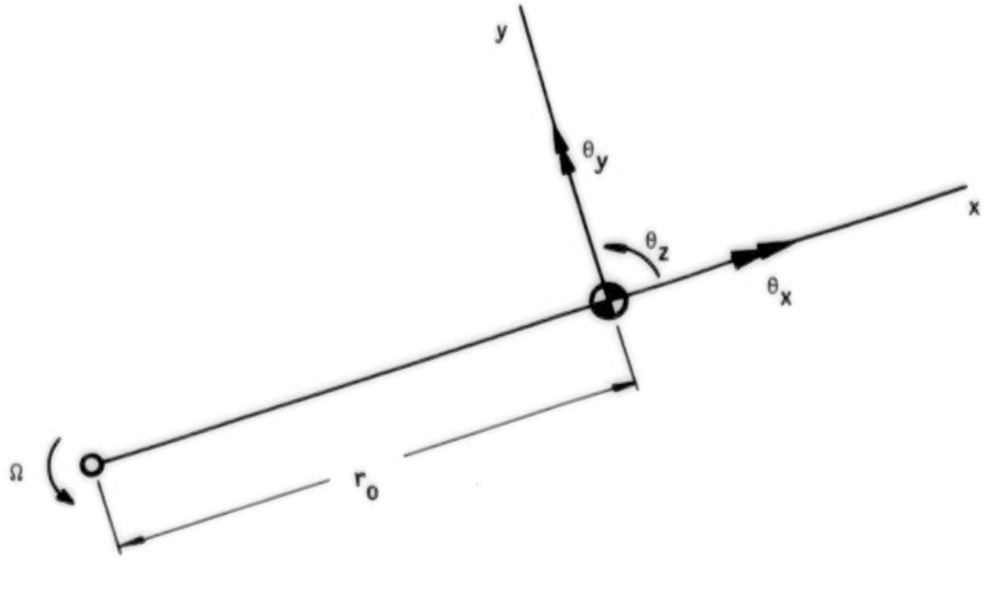
$$a_i = \vec{g} \cdot \vec{e}_i \quad , \tag{1}$$

where $\vec{g}$ is the gravity vector and $\vec{e}_i$ is a unit vector in the direction of u_i. For rotations, a_i is zero. The gravity load is then computed from

$$\{P^{gr}\} = [M_{gg}]\{a\} \quad , \tag{2}$$

where $[M_{gg}]$ is the mass matrix referred to the u_g displacement set. It should be noted that the gravitational acceleration is not calculated at scalar points. The direction of motion at scalar points is established indirectly by constraints and by other forms of coupling with geometric grid points. The user is required to introduce gravity loads at scalar points directly.

A centrifugal force load is specified by the designation of a grid point that lies on the axis of rotation and by the components of rotational velocity in a defined coordinate system. The components of force acting on a rigid body in a centrifugal force field are most simply expressed in a Cartesian coordinate system that is centered at the center of gravity of the body with axes directed as shown below.



The components of load are

$$\{P^{cf}\} = \begin{Bmatrix} P_x \\ P_y \\ P_z \\ M_x \\ M_y \\ M_z \end{Bmatrix} = \Omega^2 \begin{Bmatrix} mr_o \\ 0 \\ 0 \\ -I_{yz} \\ I_{xz} \\ 0 \end{Bmatrix} , \tag{3}$$

where m is the mass of the body, $I_{xz} = \int \rho xz dV$, and $I_{yz} = \int \rho yz dV$. For use in the program, the components of force and moment are transferred from the center of gravity to the grid point and its local coordinate system; the elements of the mass matrix, M_{gg}, are used in the calculation of the loads. Note, however, that the mass matrix is regarded as pertaining to a set of distinct rigid bodies connected to grid points. Deviations from this viewpoint, such as the use of scalar masses or the use of mass coupling between grid points, can result in errors.

The equivalent loads due to thermal expansion are calculated by separate subroutines for each type of structural element, and are then transferred from the internal coordinates of the element to the coordinates of the surrounding grid points. The equations that define the equivalent forces and moments are derived for each element in Section 5.

The user may define temperatures by more than one method. For BARS, RØDS, and PLATES the temperature may be specified for each individual element. The temperature specification for BARS and RØDS includes the average temperature and, in the case of the BAR element, the effective transverse thermal gradient at each end. The temperature of a PLATE element can vary arbitrarily in the direction of the thickness, but it is assumed to be independent of position on the surface. For all other elements that permit thermal expansion, and for BARS, RØDS, and PLATES if their temperatures are not individually specified, the temperature is obtained by averaging the temperatures specified at the grid points to which the element is attached. Temperature-dependent thermal expansion coefficients and elastic moduli are stored in material properties tables which the user applies to each structural element by specifying the code number of its material. The average temperature of an element is used to determine its temperature-dependent material properties.

Enforced axial deformations can be applied to the one-dimensional elements (BARS and RODS). They are useful in the simulation of misfit and misalignment in engineering structures. As in the case of thermal expansion, the equivalent loads are calculated by separate subroutines for each type of structural element. In the case of a bar, for example, the equivalent loads placed at the ends are equal to $EA\delta u/\ell$ where E is the modulus, A is the cross-sectional area, δu is the enforced expansion, and ℓ is the length of the bar.

Enforced displacements at grid points are discussed in connection with single point constraints, Section 3.5.2.

3.6.2 Reduction of Load Vectors to Final Form and Solution for Displacements

The operations by which structural matrices and load vectors are reduced from the u_g set to the u_ℓ set have been described in Section 3.5. In the program, the reduction of load vectors to final form is performed in a single module, (SSG2). The operations are summarized below.

1. Partition the load vector, $\{P_g\}$, whose generation is described in the preceding subsection, according to the set of coordinates, u_m, that are restrained by multipoint constraints, and the set, u_n, that are not.

$$\{P_g\} = \left\{\begin{array}{c} \bar{P}_n \\ -- \\ P_m \end{array}\right\} . \tag{4}$$

2. Eliminate multipoint constraints.

$$\{P_n\} = \{\bar{P}_n\} + [G_m]^T\{P_m\} . \tag{5}$$

3. Partition $\{P_n\}$ according to the set of coordinates, u_s, that are restrained by single point constraints and the set, u_f, of free coordinates.

$$\{P_n\} = \left\{\begin{array}{c} \bar{P}_f \\ -- \\ P_s \end{array}\right\} . \tag{6}$$

4. Eliminate single point constraints.

$$\{P_f\} = \{\bar{P}_f\} - [K_{fs}]\{Y_s\} . \tag{7}$$

5. Partition $\{P_f\}$ according to the optional partition scheme described in Section 3.5.3.

$$\{P_f\} = \begin{Bmatrix} \bar{P}_a \\ -- \\ P_o \end{Bmatrix} . \tag{8}$$

6. Eliminate the set of omitted coordinates, u_o.

$$\{P_a\} = \{\bar{P}_a\} + [G_o]^T\{P_o\} . \tag{9}$$

7. Partition $\{P_a\}$ according to the set of coordinates, u_r, that are restrained by free body reactions, and the set, u_ℓ, that are not.

$$\{P_a\} = \begin{Bmatrix} P_\ell \\ -- \\ P_r \end{Bmatrix} . \tag{10}$$

$\{P_\ell\}$ is the load vector in final form.

In the program the displacement vector set, u_ℓ, is obtained from solution of the equation

$$[K_{\ell\ell}]\{u_\ell\} = \{P_\ell\} , \tag{11}$$

in a separate module, (SSG3). It will be recalled, Section 3.5.5, that the triangular factors of $[K_{\ell\ell}]$ were previously computed in order to form the rigid body matrix, [D]. The operations performed in SSG3 are the forward and backward passes through the triangular factors of $[K_{\ell\ell}]$ (see Section 2.3) for each loading condition.

The vector set, u_o^o, that describes displacements of the omitted set relative to the remaining set (see Section 3.5.3) is also obtained in SSG3 from solution of the equation

$$[K_{oo}]\{u_o^o\} = \{P_o\} . \tag{12}$$

The triangular factors of $[K_{oo}]$ were previously computed in order to form $[G_o]$.

Double precision arithmetic is used in the formation and triangular decomposition of structural matrices, so that significant error due to the accumulation of round-off is regarded as unlikely. Such errors can occur, however, in exceptionally ill-conditioned problems (see Section

15.1). A test is provided in NASTRAN on the solution of Equations (11) and (12) that will indicate the presence of trouble to the user.

In the test a residual load vector is first obtained from

$$\{\delta P_\ell\} = \{P_\ell\} - [K_{\ell\ell}]\{u_\ell\} \tag{13}$$

The work done by the residual load vector is then compared with the work done by the applied load vector in the residual energy criterion,

$$\varepsilon_e = \frac{\{u_\ell\}^T\{\delta P_\ell\}}{\{u_\ell\}^T\{P_\ell\}} \; . \tag{14}$$

Iterative improvement, such as might be obtained by computing second and higher order residual load vectors,

$$\{\delta^{(n)}P\} = \{\delta^{(n-1)}P\} - [K]\{\delta^{(n-1)}u\} \quad , \tag{15}$$

is not attempted. The gain in accuracy from iterative improvement is largely illusory because errors made in the formation of [K], which are of the same order as those made in the triangular decomposition of [K], are uncorrected. This matter is discussed more fully in Section 15.1.

3.6.3 Inertia Relief

When a free body is subjected to loads that are not in equilibrium, the body is accelerated in its rigid body (or more generally, free body) modes. If the time rate of change of the applied loads is small compared to the frequency of the lowest elastic mode of the system, an approximate state of equilibrium exists between the applied loads and the inertia forces due to acceleration. Stresses in the body may be computed, in this case, from an applied load distribution that includes the inertia forces. The term "inertia relief" is applied to the effect that the inertia forces have on the stresses. In order for an "effect" to be defined, a condition in which the effect does not exist must be imagined. In the case of inertia relief, the "effect-free" condition is one in which the free body is restrained by determinate supports. The choice of support points is arbitrary, but usually corresponds to a natural or customary location (e.g. the intersection between wing and fuselage of an aircraft). Although the condition including inertia effects is the correct solution, the analyst may also be interested in the results for the supported condition.

The general procedure for including inertia relief in static analysis is as follows:

1. Select, from the displacement vector $\{u_a\}$, a subset $\{u_r\}$ of determinate support points as has been discussed in Section 3.5.5.
2. Find the accelerations $\{\ddot{u}_r\}$ due to the applied loads $\{P_a\}$. This requires evaluation of the rigid body mass matrix $[m_r]$ referred to points $\{u_r\}$.
3. Calculate accelerations at all other points $\{u_\ell\}$ and the corresponding inertia forces.
4. Add the inertia force vector to the applied load vector and solve for the displacements $\{u_\ell\}$ while the structure is rigidly restrained at points, $\{u_r\}$. The forces of reaction will be zero.

The equations of motion for the body, expressed in terms of the displacements, u_r, can be written

$$[m_r]\{\ddot{u}_r\} = \{\overline{P}_r\} = -\{q_r\} \quad . \qquad (16)$$

$[m_r]$ is the mass matrix reduced to the u_r coordinates. It is evaluated from partitions of the $[M_{aa}]$ matrix by means of Equation 46 of Section 3.5.5. $\{\overline{P}_r\}$ is the applied load vector reduced to the u_r coordinates. It is numerically equal to $-\{q_r\}$, the set of determinate reactions, evaluated in Equation 42 of Section 3.5.5.

Solution of Equation 16 gives

$$\{\ddot{u}_r\} = -[m_r]^{-1}\{q_r\} \quad . \qquad (17)$$

The accelerations of the remaining points $\{u_\ell\}$, assuming uniform acceleration as a rigid body, are obtained from Equation 44 of Section 3.5.5,

$$\{\ddot{u}_\ell\} = [D]\{\ddot{u}_r\} \quad . \qquad (18)$$

The inertia forces acting on the u_ℓ coordinates are, utilizing the partitions of the $[M_{aa}]$ matrix shown in Equation 45 of Section 3.5.5.

$$\begin{aligned}\{P_\ell^i\} &= -[M_{\ell\ell}]\{\ddot{u}_\ell\} - [M_{\ell r}]\{\ddot{u}_r\} \\ &= [M_{\ell\ell}D + M_{\ell r}][m_r]^{-1}\{q_r\} \quad . \qquad (19)\end{aligned}$$

The vector $\{P_\ell^i\}$ is added to the applied load vector $\{P_\ell\}$ in problems where the inertia relief effect is included. Since $[m_r]$ is usually of small order, its inversion is not troublesome.

The inertia relief effect is also included in the calculation of the displacement set, u_o^o, that expresses the motions of the omitted coordinates, u_o, relative to the u_a coordinates. The inertia force vector for the omitted coordinates is

$$\{P_o^i\} = \{-M_{oo}\ddot{u}_o - M_{ao}{}^T\ddot{u}_a\} \quad . \tag{20}$$

Now, if acceleration as a rigid body is assumed,

$$\{\ddot{u}_a\} = \left\{\frac{\ddot{u}_\ell}{\ddot{u}_r}\right\} = \left[\frac{D}{I}\right]\{\ddot{u}_r\} = -\left[\frac{D}{I}\right][m_r]^{-1}\{q_r\} \ , \tag{21}$$

and

$$\{\ddot{u}_o\} = [G_o]\{\ddot{u}_a\} \ . \tag{22}$$

Thus, the inertia force vector for the omitted coordinates is

$$\{P_o^i\} = [M_{oo}G_o + M_{ao}{}^T]\left[\frac{D}{I}\right][m_r]^{-1}\{q_r\} \ , \tag{23}$$

which should be added to $\{P_o\}$ in Equation 12.

ORIGINAL PAGE IS
OF POOR QUALITY

3.7 DATA RECOVERY

Data recovery in static analysis by the displacement method is performed in two steps:

1. Recovery of displacement sets that were eliminated during the reduction of the stiffness matrix to final form, resulting in the formation of the complete grid point displacement vector, u_g.
2. Recovery of internal forces and stresses in structural elements, using the grid point displacement vector, u_g, to define the displacements at the corners of each element. Margins of safety are also calculated. Separate subroutines are used for each type of element.

The above steps are discussed in separate subsections.

3.7.1 Recovery of Displacements

Solutions for the vector sets, u_ℓ and u_o^o, are discussed in Section 3.6.2. The remaining operations required to recover the complete grid point displacement vector, u_g, are as follows:

1. Merge u_r, whose elements are all zero in static analysis, with u_ℓ to form u_a.

$$\left\{\frac{u_\ell}{u_r}\right\} \rightarrow \{u_a\} \quad . \tag{1}$$

2. Recover the omitted coordinates, u_o .

$$\{u_o\} = [G_o]\{u_a\} + \{u_o^o\} \ . \tag{2}$$

3. Merge u_o and u_a to form the vectors of free coordinates, u_f.

$$\left\{\frac{u_a}{u_o}\right\} \rightarrow \{u_f\}. \tag{3}$$

4. Evaluate the single point constraint set, u_s.

$$\{u_s\} = \{Y_s\} \quad . \tag{4}$$

   $\{Y_s\}$ is the vector of enforced displacements.

5. Merge u_f and u_s to form u_n.

116

$$\left\{\begin{array}{c} u_f \\ \hdashline u_s \end{array}\right\} \rightarrow \{u_n\} \quad . \tag{5}$$

6. Recover the multipoint constraint (and rigid element) set, u_m.

$$\{u_m\} = [G_m]\{u_n\} \quad . \tag{6}$$

The forces of constraint due to multipoint constraints and rigid elements are not available as part of the standard calculations in the program, but may be obtained in statics and real eigenvalue problems by ALTERing in the module EQMCK and requesting these forces in the Case Control Deck. The constraint forces are given by

$$\{q_g^m\} = \left\{\begin{array}{c} q_n^m \\ \hdashline q_m^m \end{array}\right\} \quad , \tag{6a}$$

where

$$\{q_m^m\} = -\{P_m\} + [K_{nm}]^T\{u_n\} + [K_{mm}]\{u_m\} \quad , \tag{6b}$$

and

$$\{q_n^m\} = -[G_m]^T\{q_m^m\} \quad . \tag{6c}$$

7. Merge u_n and u_m to form u_g.

$$\left\{\begin{array}{c} u_n \\ \hdashline u_m \end{array}\right\} \rightarrow \{u_g\} \quad . \tag{7}$$

The matrices $[G_o]$ and $[G_m]$, used in the data recovery process, were generated during the reduction of the structural matrices to final form and were placed in auxiliary storage.

A miscellaneous task that is performed in the same module that recovers u_g is the recovery of the single point forces of constraint,

$$\{q_s\} = -\{P_s\} + [K_{fs}]^T\{u_f\} + [K_{ss}]\{u_s\} \quad . \tag{8}$$

3.7.2 Recovery of Stress Data

Internal forces and stresses in structural elements (as well as strains and curvatures in certain elements) are calculated from knowledge of the displacements at the grid points bounding the element and the physical parameters of the element, including geometric properties, elastic properties, and temperature. The equations by which internal forces and stresses are calculated are contained in a separate subroutine for each type of element. They are discussed in Section 5.

In the calculation procedure, the stress recovery parameters for as many elements as possible are placed in the high speed memory. The stresses are computed from the u_g vector for the first loading condition, and are placed in peripheral storage. The u_g vectors for other loading conditions are then processed sequentially. The procedure is repeated for additional structural elements (if any) that could not be stored initially. The procedure that has been described makes minimum use of INPUT/OUTPUT data transfers. For most elements, I/O transfers are the limiting factor on computational speed in stress data recovery.

A number of different kinds of stress data are available for each type of element. With the BAR element, for example, the user can request any or all of the following:

- Bending moments at both ends in two planes.
- Transverse shear forces in two planes.
- Axial force.
- Torque.
- The average axial stress.
- The extensional stress due to bending at four points on the cross-section at both ends. The points are specified by the user.
- The maximum and minimum extensional stresses at both ends.
- Margins of safety in tension and compression for the whole element.
- Number of digits of accuracy for the element forces.

The capability exists for certain elements to indicate the number of significant digits of accuracy for stresses and forces. Numerical problems may exist in using the differences in displacements to calculate stresses and forces. For instance, displacements may be large relative to the overall structure, yet the differences may be small relative to an individual element.

The precision is calculated by

$$h_i = -\log_{10}\left(\frac{\left|\sum_j S_{ij}u_j\right|}{\sum_j \left(|S_{ij}|\cdot|u_j|\right)}\right) = -\log_{10}\frac{|\sigma_i|}{D_i} \quad , \qquad (8a)$$

where σ_i is the specific stress component; D_i is calculated using absolute values for the stress equations, S_{ij} are the stress matrix components and u_j are the displacement components. The number of significant digits, N_{di}, is calculated by

$$N_{di} = \log_{10}(2^p) - h_i \quad , \tag{8b}$$

where p is approximately the number of bits in the mantissa for the machine being used.

The results are printed directly in a format unique to the element type instead of by the output file processor. Only the results for those elements with an entry less than the user-defined precision are printed.

3.7.3 Grid Point Force Balance and Element Strain Energy Distribution

The new method of element matrix generation introduced in NASTRAN Level 16 includes the feature that the elastic stiffness matrix for each element is individually saved in peripheral storage. This feature makes practical a number of capabilities which would otherwise be prohibitively expensive, including the determination of force balances at grid points and the calculation of the strain energy distribution by elements throughout the structure.

The vector of elastic forces exerted by a structural element on its connecting grid points is related to the displacements at these points by

$$\{F_e\} = -[K_{ee}]\{u_e\} \quad , \tag{9}$$

where $\{u_e\}$ is the subset of the global degrees of freedom, $\{u_g\}$, to which the element is connected. The matrix $[K_{ee}]$ is computed by module EMG and stored.

The grid point force balances computed by NASTRAN include the force and moment contributions in the global coordinate system of element elastic forces computed by Equation 9, applied loads and single point forces of constraint. They do not include, at present, forces due to differential stiffness, multipoint constraints, general elements, or any dynamic effects. NASTRAN prints the individual contributions of the former effects and their sum at each grid point. If none of the latter effects are present, the sum is due to round-off error. The sum is not the same as the residual used in the ε test (see Section 2.3) because it is calculated at a different time in a different manner.

The strain energy within an individual element is

$$W_e = \frac{1}{2}\{F_e\}^T\{u_e\} \quad . \tag{10}$$

The user can request a list of the percentages of the total strain energy stored in each element.

3.7.4 Overall Equilibrium Check

The quality or closure in the solution of a statics problem or the net momentum in a real eigenvalue problem is indicated by the equilibrium of the forces about the basic coordinate system origin or any selected grid point. The forces on the structure are obtained in module EQMCK from the following sources:

1. Directly applied loads: $\{P_g\}$
2. Forces of single-point constraint: $\{q_s\}$
3. Forces due to multipoint constraints and rigid elements: $\{q_g^m\}$
4. Forces of reaction: $\{q_r\}$

The rigid body transformation matrix $[D]^T$ about the reference point is the same as for the grid point weight generator module (see description in Section 4.29.7 of the Programmer's Manual). Thus, the net force and moment vector may be obtained by:

$$\left.\begin{aligned} \{S_p\} &= [D]^T\{P_g\} \\ \{S_g\} &= [D]^T\{q_g\} \\ \{S_g^m\} &= [D]^T\{q_g^m\} \end{aligned}\right\} \quad , \tag{11}$$

resulting in the summation,

$$\{S_t\} = \{S_p + S_g + S_g^m\} = [D]^T\{P_g + q_g + q_g^m\} \quad , \tag{12}$$

where q_g is the union of the forces q_s and q_r. Note that the effects of scalar points are ignored in the [D] matrix, and the forces resulting from "grounded" scalar springs and general elements are not calculated. Each of the {S} vectors in Equation 11, consisting of three forces and three moments, is output separately along with the net equilibrium forces and moments from Equation 12.

The forces due to multipoint constraints and rigid elements, q_g^m, as calculated by Equations 6a, 6b, and 6c, may be requested as an output file for printing by the output file processor.

3.8 PIECEWISE LINEAR ANALYSIS

The Piecewise Linear Analysis option of NASTRAN is used to solve problems in material plasticity. The load is applied in increments such that the stiffness properties can be assumed to be constant over each increment. The stiffness matrix for each increment is dependent on the current states of stress in the structural elements. The increments in displacements and stresses are accumulated to produce the final, nonlinear results. Since the algorithm assumes linearity between sequential loads, the results will depend on the user's choice of load increments. When the user selects large load increments and the material properties are changing rapidly, the results may be unacceptably inaccurate. If small load increments are used when the structure is nearly linear the solution will be very accurate but relatively costly.

3.8.1 Limitations and Available Options

The nonlinearity of a structural element is defined by the material used by the element. Any isotropic material may be made nonlinear by including a stress-strain table defining its extension test characteristics.

The stress-strain table must define a nondecreasing sequence of both stresses and strains. Because the stiffness matrix for the first load increment uses the elastic material coefficients, the initial slope should correspond to the defined Young's Modulus, E.

The nonlinear effects depend on the element type. The elements which utilize the plastic material properties are described in Section 3.8.4.

Linear elements and materials may be used in any combination with the nonlinear elements. Elements with low stress states may be included in this category by providing them with "linear" material properties even though their actual properties are decidedly nonlinear at high stress levels. Linear elements are used in a more efficient manner than the nonlinear elements.

All static load options except temperature and enforced element deformation are allowed with piecewise linear analysis. The reason for the exceptions is that the equivalent grid point loads depend on the stiffness of the structure and hence on the sequence of their application. For example, changing temperature after a load is applied gives different results than changing temperature before the load is applied.

All statics constraint options are available including enforced displacement at grid points. The use of enforced deformation in combination with applied loads has the ambiguity discussed

PAGE 121 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

above. In the program enforced displacements are increased simultaneously with the other loads.

No protective steps are taken to prevent the attempted decomposition of a singular stiffness matrix. If the structure fails by buckling or yielding, a solution is still attempted and the results will be obviously erroneous.

3.8.2 Overall Solution Logic

Although the Piecewise Linear Analysis rigid format uses many of the statics analysis modules, the path through the various operations is substantially different. A summary flow diagram is given in Figure 1. The various steps are given numbers corresponding to the explanations below:

1. The normal statics analysis "front end" is used to generate the grid point, element, and loading tables. The stiffness matrix (and the mass matrix for gravity loads) is generated in the normal manner using the moduli of elasticity given with the materials.

2. The element tables are separated into linear and nonlinear elements. The program recognizes a nonlinear element as one that has a stress-strain table referred to by its material. The linear elements are used to generate a linear stiffness matrix, $[K^{\ell}_{gg}]$. This matrix will not change with loading changes.

3. The load vector for the whole structure, $\{P_g\}$, is generated by the normal methods except that loads due to temperatures and enforced element deformations are ignored. The constrained points are also identified in this stage.

4. The "current" stiffness matrix is initially the linear elastic matrix; for subsequent load increments the matrix is changed as shown in step 8. The constraints are applied to the matrix in the normal sequence to produce the $[K_{\ell\ell}]$, $[K_{fs}]$, $[K_{ss}]$, and $[G_o]$ matrices. The $[K_{\ell\ell}]$ matrix is decomposed to produce the triangular matrices $[U_{\ell\ell}]$ and $[L_{\ell\ell}]$.

   In a similar manner the applied loads, including enforced displacements at grid points, are modified by the constraints to produce a load vector for the independent coordinates, $\{P_\ell\}$. The current load increment is:

$$\{\Delta P_\ell{}^i\} = (\alpha_i - \alpha_{i-1})\{P_\ell\} = \Delta\alpha_i\{P_\ell\} \qquad i = 1,2,\ldots n \;, \qquad (1)$$

   where α_1, α_2, ... n are a set of load level factors provided by the user.

123

5. The incremental displacements are generated using the current stiffness matrix and the current load vector increment. The dependent displacements are recovered in the normal manner and merged to produce the increments for all degrees of freedom, $\{\Delta u_g^{\,i}\}$. The incremental forces of single-point constraint, $\{\Delta q_s^{\,i}\}$, are also recovered. The increments are added to the previous vectors to produce the current vectors

$$\{u_g^{\,i}\} = \{u_g^{\,i-1}\} + \{\Delta u_g^{\,i}\} \quad , \tag{2}$$

$$\{q_s^{\,i}\} = \{q_s^{\,i-1}\} + \{\Delta q_s^{\,i}\} \quad . \tag{3}$$

6. The total nonlinear element stresses are calculated for output within the loop so that the user may have some useful information in case of an unscheduled exit before the end of the calculation. The method of calculating stresses is given in Section 3.8.4.

7. The stiffness matrix for the nonlinear elements, $[K_{gg}^{\,n}]$, is generated six columns at a time for all nonlinear elements connected to a grid point. The table of element connections and properties is appended to include the current stress and strain values. The modulus of elasticity is calculated from the slope of the stress-strain curve as explained below.

8. The nonlinear element stiffness matrix, generated in step 7, is added to the linear element stiffness matrix, generated in step 2, to produce a new stiffness matrix. The next pass through the loop will reflect the new stress state of the structure.

9. When the results for all load increments have been produced, the data are output. Stresses for the linear elements are calculated directly from the total displacement vector.

3.8.3 Piecewise Linear Stress-Strain Functions

In order to simplify input to the program, a single type of plastic material table is used. A stress-strain tabular function is input for each nonlinear material. Only certain types of elements may use the nonlinear tables.

In calculating the current elastic constants of a plastic element, an approximation to the slope of the stress-strain function is used. Because the elastic constants are to be used for the interval between the present load and the next load, an extrapolation of current information is required. A linear extrapolation is used to estimate the strain due to the next load increment as shown in Figure 3. The current strain increment, $\Delta\varepsilon_i$, is computed from the current displacement increment $\Delta u_g{}^i$ by separate subroutines for each type of element. The next strain increment, $\Delta\varepsilon_{i+1}$, is estimated by linear extrapolation,

$$\Delta\varepsilon_{i+1} = \varepsilon_{i+1} - \varepsilon_i = A(\Delta\alpha_{i+1}) \ , \qquad (4)$$

where A is obtained by the curve fit through two previously computed points, ε_i, and ε_{i-1}.

The linear elastic modulus, E_0, is used for the first increment. For all succeeding increments

$$E_{i+1} = \frac{\sigma_{i+1} - \sigma_i}{\varepsilon_{i+1} - \varepsilon_i} \ , \qquad (5)$$

where σ_i and σ_{i+1} are obtained from points on the stress-strain curve, Figure 2, corresponding to ε_i and ε_{i+1}.

The actual strain components used above depend on the element type. A brief description of the elements used in piecewise linear analysis is given below.

3.8.4 Element Algorithms for Piecewise Linear Analysis

3.8.4.1 ROD, TUBE, and BAR Elements

The plasticity of these elements is assumed to depend on the state of extensional stress only. Bending and twisting stresses are ignored in the determination of the effective elastic constants. If bending stresses are important, the bar may be represented in NASTRAN as a built-up structure composed of rods, shear panels and/or plates. The estimated next extensional strain is:

$$\varepsilon_{i+1} = \varepsilon_i + \gamma_i \Delta\varepsilon_i \ , \qquad (6)$$

where the coefficient γ_i is the ratio of load increments

$$\gamma_i = \frac{\alpha_{i+1} - \alpha_i}{\alpha_i - \alpha_{i-1}} = \frac{\Delta\alpha_{i+1}}{\Delta\alpha_i} \ . \qquad (7)$$

The elastic constants for the new stiffness matrices and the next stress calculations are:

$$E_{i+1} = \frac{\sigma_{i+1} - \sigma_i}{\varepsilon_{i+1} - \varepsilon_i} \ , \quad G_{i+1} = \frac{G_o}{E_o} E_{i+1} \ , \tag{8}$$

where G_o and E_o are the elastic shear and extensional moduli given with the material. These constants are used in the calculation of extensional, bending, twisting, and transverse shear stiffnesses in the next increment.

3.8.4.2 Plate Elements

The in-plane stresses of plate elements are used to calculate the elastic properties for in-plane deformations. They are also used to calculate the elastic properties for bending and transverse shear, except in the case of those plate elements where the bending and transverse shear material are different from the membrane material.

Plastic, rather than nonlinear elastic, behavior is assumed. The theoretical basis of two dimensional plastic deformation as used in NASTRAN is that developed by Swedlow (Reference 1). Only a summary of the theory will be presented here. In the development, a unique relationship between the octahedral stress, τ_o, and the plastic octahedral strain, ε_o^P, is assumed to exist. The total strain components (ε_x, ε_y, ε_z, and γ_{xy}) are composed of the elastic, recoverable deformations and the plastic portions (ε_x^P, ε_y^P, ε_z^P, and γ_{xy}^P). The rates of plastic flow, ($\dot{\varepsilon}_x^P$, etc.), are independent of a time scale and are simply used for convenience instead of incremental values.

The definitions of the octahedral stress and the octrahedral plastic strain rate are:

$$\tau_o = \frac{1}{3}\sqrt{2(S_{11}{}^2 + 2S_{12}{}^2 + S_{22}{}^2 + S_{33}{}^2)} \ , \tag{9}$$

$$\dot{\varepsilon}_o^P = \sqrt{[(\dot{\varepsilon}_{11}{}^P)^2 + 2(\dot{\varepsilon}_{12}{}^P)^2 + (\dot{\varepsilon}_{22}{}^P)^2 + (\dot{\varepsilon}_{33}{}^P)^2]/3} \ , \tag{10}$$

where

$$\left.\begin{aligned} S_{11} &= \tfrac{1}{3}(2\sigma_x - \sigma_y) & \epsilon_{11}{}^P &= \epsilon_x{}^P \\ S_{12} &= \tau_{xy} & \epsilon_{12}{}^P &= \tfrac{1}{2}\gamma_{xy}{}^P \\ S_{22} &= \tfrac{1}{3}(2\sigma_y - \sigma_x) & \epsilon_{22}{}^P &= \epsilon_y{}^P \\ S_{33} &= -\tfrac{1}{3}(\sigma_x + \sigma_y) & \epsilon_{33}{}^P &= \epsilon_z{}^P \end{aligned}\right\} \qquad (11)$$

The S_{ij} array is called the "deviator" of the stress tensor. σ_x, σ_y, and σ_{xy} are the Cartesian stresses. The basic Prandtl-Reuss flow rule is:

$$\dot{\epsilon}_{ij}{}^P = \dot{\lambda}\, S_{ij} \quad , \qquad (12)$$

where $\dot{\lambda}$ is a flow rate parameter.

$\dot{\lambda}$ may be derived by multiplying Equation 12 by itself according to the rules of tensor analysis to produce a scalar equation. The result is:

$$\dot{\lambda} = \frac{\dot{\epsilon}_o{}^P}{\tau_o} \quad . \qquad (13)$$

Another basic assumption is that the material yields according to its octahedral stress and strain. In other words, there exists a function, $M_T(\tau_o)$, such that

$$\frac{\dot{\tau}_o}{\dot{\epsilon}_o{}^P} = 2M_T(\tau_o) \quad . \qquad (14)$$

Combining Equations 12, 13, and 14 we obtain

$$\dot{\epsilon}_{ij}{}^P = \frac{\dot{\tau}_o\, S_{ij}}{2\tau_o M_T(\tau_o)} \quad . \qquad (15)$$

Taking the derivatives of Equation 9 we obtain:

$$\dot{\tau}_o = \frac{1}{3\tau_o}\left(S_{11}\dot{S}_{11} + 2S_{12}\dot{S}_{12} + S_{22}\dot{S}_{22} + S_{33}\dot{S}_{33}\right) \quad . \qquad (16)$$

Substituting the relations between the S_{ij} and the σ_{ij} terms we obtain:

$$\dot{\tau}_o = \frac{1}{3\tau_o}\left[S_{11}\dot{\sigma}_x + S_{22}\dot{\sigma}_y + 2S_{12}\dot{\tau}_{xy}\right] \quad . \tag{17}$$

The matrix relationship for the plastic flow is formed from Equations 11, 15, and 17:

$$\begin{Bmatrix} \dot{\varepsilon}_x{}^P \\ \dot{\varepsilon}_y{}^P \\ \dot{\gamma}_{xy}{}^P \end{Bmatrix} = \frac{1}{6\tau_o{}^2\, M_T(\tau_o)} \begin{bmatrix} S_{11}{}^2 & S_{11}S_{22} & 2S_{11}S_{12} \\ S_{11}S_{22} & S_{22}{}^2 & 2S_{22}S_{12} \\ 2S_{11}S_{12} & 2S_{22}S_{12} & 4S_{12}{}^2 \end{bmatrix} \begin{Bmatrix} \dot{\sigma}_x \\ \dot{\sigma}_y \\ \dot{\tau}_{xy} \end{Bmatrix} \tag{18}$$

For piecewise linear analysis this matrix, $[D^P]$, is assumed constant for a given load increment. The time derivatives are replaced with incremental values. The total strain increments, obtained by adding the plastic and linear elastic parts, are:

$$\{\Delta\varepsilon\} = ([D^P] + [G]^{-1})\{\Delta\sigma\} = [G_p]^{-1}\{\Delta\sigma\} \quad , \tag{19}$$

where $[G]$ is the normal elastic material matrix and $[G_p]$ is the equivalent plastic material matrix.

A further relationship to be derived is that of the plastic modulus, $M_T(\tau_o)$, versus the slope of a normal stress-strain curve. If a specimen is under an axial load, its stress and strain values are:

$$\left.\begin{aligned} \sigma_x &= \sigma_a \\ \sigma_y &= \sigma_z = \tau_{xy} = \tau_{xz} = \tau_{yz} = 0 \\ \varepsilon_x &= \varepsilon_x{}^P + \frac{\sigma_a}{E_o} \\ \varepsilon_y &= \varepsilon_y{}^P - \frac{\nu\sigma_a}{E_o} \\ \varepsilon_z &= \varepsilon_z{}^P - \frac{\nu\sigma_a}{E_o} \\ \gamma_{xy} &= \gamma_{yz} = \gamma_{xz} = 0 \end{aligned}\right\} \tag{20}$$

where E_o and ν are the elastic modulus and Poisson's ratio for the elastic part of the stress-strain curve.

Because of noncompressibility the plastic strains are $\varepsilon_y{}^P = \varepsilon_z{}^P = -\frac{1}{2}\varepsilon_x{}^P$.

The stress-strain relationship given in the table is:

$$\left.\begin{aligned} \sigma_x &= F(\varepsilon_x) \quad , \\ \text{and} \qquad \frac{\Delta\sigma_x}{\Delta\varepsilon_x} &= F'(\varepsilon_x) \quad . \end{aligned}\right\} \tag{21}$$

The octahedral stresses and strains may be determined from Equations 9, 10, and 11.

$$\begin{aligned} \tau_o &= \frac{\sqrt{2}}{3}\sigma_a \quad , \\ \dot{\varepsilon}_o{}^P &= \frac{\dot{\varepsilon}_x{}^P}{\sqrt{2}} \quad . \end{aligned} \tag{22}$$

The tabular relation for octahedral stress is, therefore:

$$\tau_o = \frac{\sqrt{2}}{3} F(\varepsilon_x) \quad . \tag{23}$$

The slope relation is from Equations 22 and 23:

$$\Delta\tau_o = \frac{\sqrt{2}}{3} F'(\varepsilon_x)\, \Delta\varepsilon_x \quad , \tag{24}$$

where from Equations 20 and 22:

$$\Delta\varepsilon_x = \sqrt{2}\, \Delta\varepsilon_o{}^P + \frac{3\Delta\tau_o}{\sqrt{2}\, E_o} \quad . \tag{25}$$

The octahedral plastic strain-to-octrahedral stress function obtained by solving Equations 24 and 25, and substituting into Equation 24, is:

$$\frac{1}{2M_T(\tau_o)} = \frac{\Delta\varepsilon_o{}^P}{\Delta\tau_o} = \left(\frac{3}{2}\right)\frac{E_o - E^i(\varepsilon_x)}{E_o E^i(\varepsilon_x)} \quad , \tag{26}$$

where $E^i = F'$ is the approximate slope of the stress-strain curve at each increment.

In applying the theory the following steps are taken:

1. The strain increments, $\{\Delta\varepsilon^i\}$, are determined from the incremental displacement vector and the geometry of the element. Using the material plastic matrix from the preceeding step $[G_p^{i-1}]$, the new stresses are:

$$\{\sigma^i\} = \{\sigma^{i-1}\} + [G_p^{i-1}]\{\Delta\varepsilon^i\} \; . \tag{27}$$

2. Using Equations 9, 11, and 22 the new octahedral stress, τ_o^i, and its unidirectional equivalent, σ_a^i, are calculated. σ_a^i is used with the stress-strain table to determine a strain ε_a^i. Using the previous strain ε_a^{i-1}, a new strain, ε_a^{i+1}, is estimated by linear extrapolation as in the case of extensional elements.

   In case σ_a^i exceeds the maximum tabulated value, the incremental modulus, E^i, is set equal to zero on the assumption that the element has ruptured.

3. Using the stress-strain table, the next estimated stress, σ_a^{i+1}, is found. The stress-strain slope is:

$$E^i = \frac{\sigma_a^{i+1} - \sigma_a^i}{\varepsilon_a^{i+1} - \varepsilon_a^i} \; . \tag{28}$$

4. The new stiffness matrix, $[G_p^i]$, is calculated from Equations 9, 11, 26, 18, and 19 using σ_x^i, σ_y^i, τ_{xy}^i, and E^i as input data. $[G_p^i]$ is then used in the normal stiffness matrix calculation routine.

The quadrilateral elements use extra logic since they are composed of four overlapping triangles. The primary difference is that the stress increments are averaged over the four triangles and the resulting material matrices must be treated as anisotropic and rotated into each subelement's coordinate system.

REFERENCE

1. Swedlow, J. L., "The Thickness Effect and Plastic Flow in Cracked Plates," Aerospace Research Laboratories Report No. ARL 65-216, Wright Patterson Air Force Base, Ohio, October 1965.

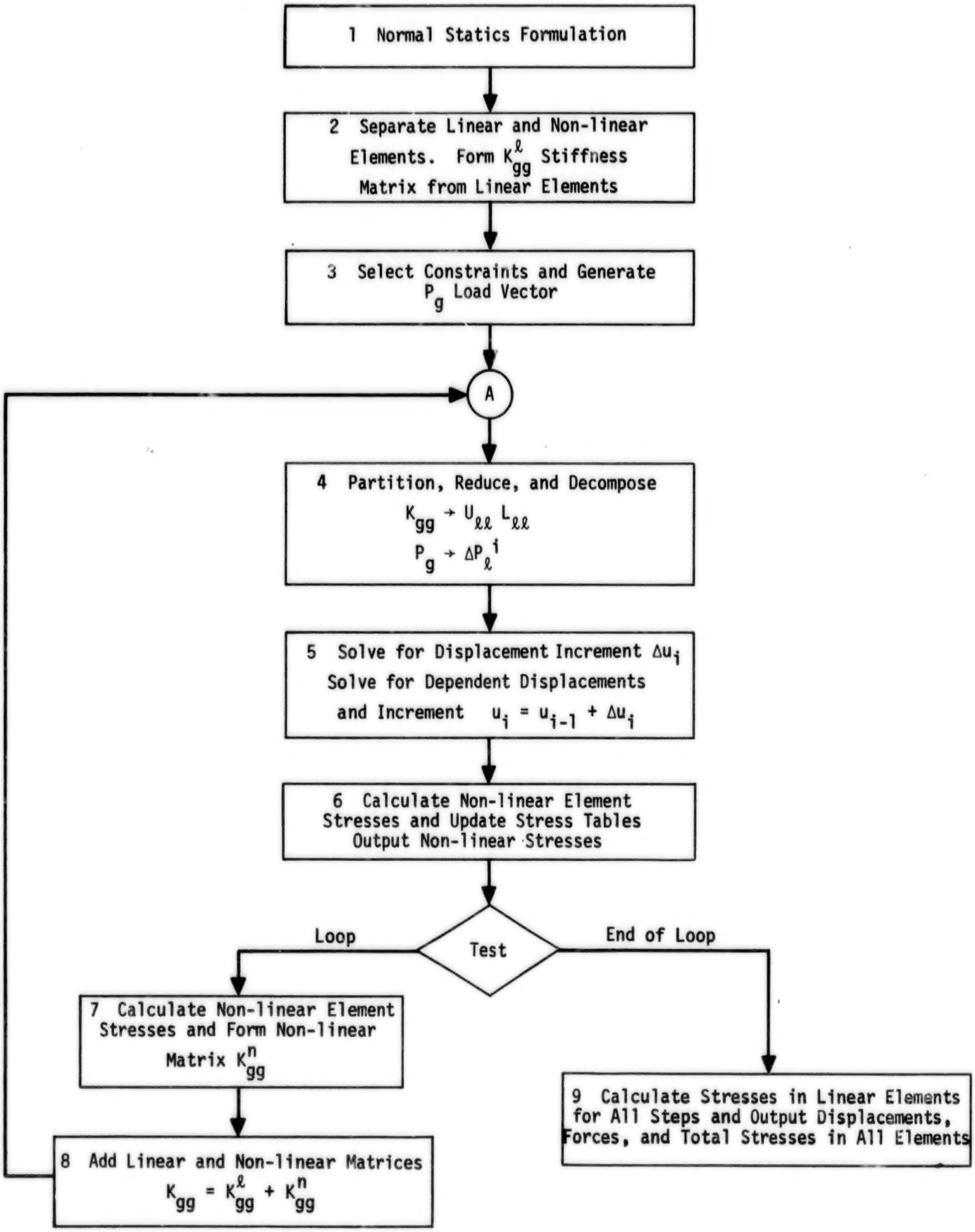

1 Normal Statics Formulation
2 Separate Linear and Non-linear Elements. Form K_{gg}^{ℓ} Stiffness Matrix from Linear Elements
3 Select Constraints and Generate P_g Load Vector
A
4 Partition, Reduce, and Decompose
$K_{gg} \rightarrow U_{\ell\ell} L_{\ell\ell}$
$P_g \rightarrow \Delta P_{\ell}^{i}$
5 Solve for Displacement Increment Δu_i
Solve for Dependent Displacements and Increment $u_i = u_{i-1} + \Delta u_i$
6 Calculate Non-linear Element Stresses and Update Stress Tables Output Non-linear Stresses
Loop
Test
End of Loop
7 Calculate Non-linear Element Stresses and Form Non-linear Matrix K_{gg}^{n}
8 Add Linear and Non-linear Matrices
$K_{gg} = K_{gg}^{\ell} + K_{gg}^{n}$
9 Calculate Stresses in Linear Elements for All Steps and Output Displacements, Forces, and Total Stresses in All Elements

Figure 1. Piecewise linear flow diagram.

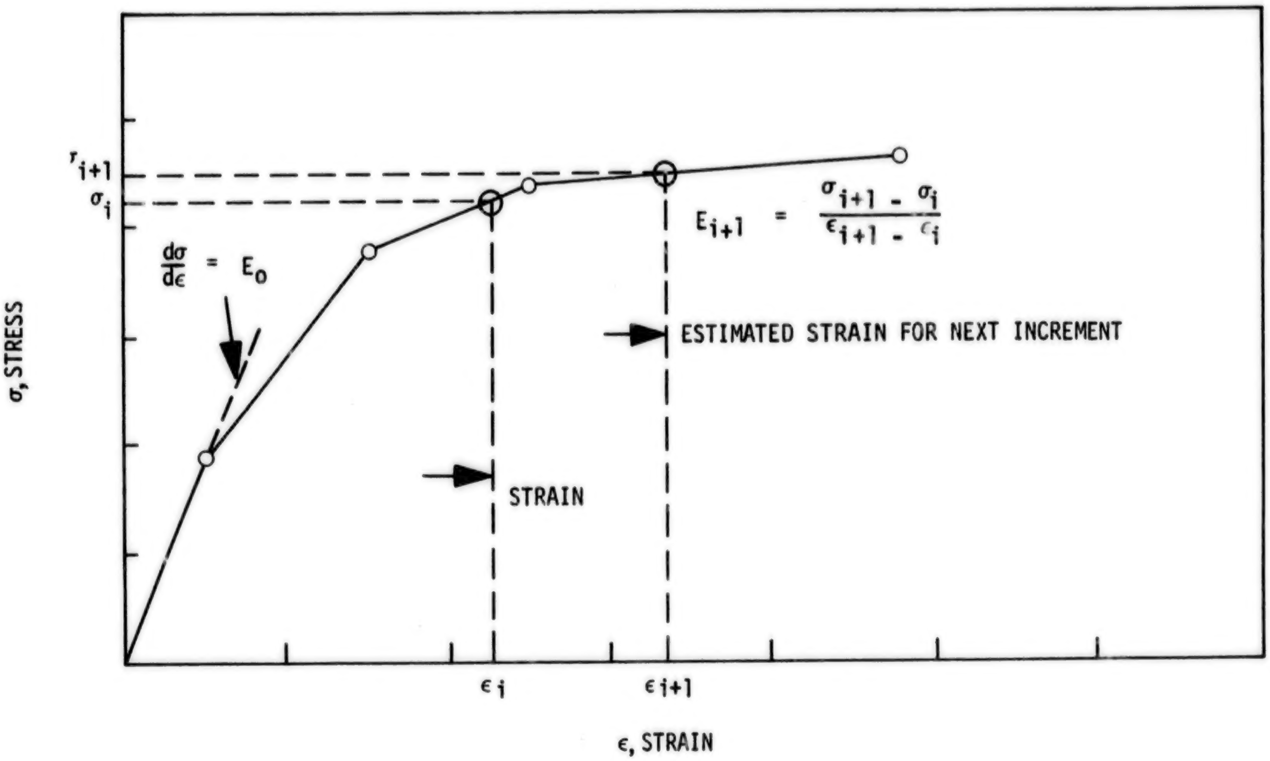


Figure 2. Determination of elastic modulus from tabular stress-strain curve defined by user.

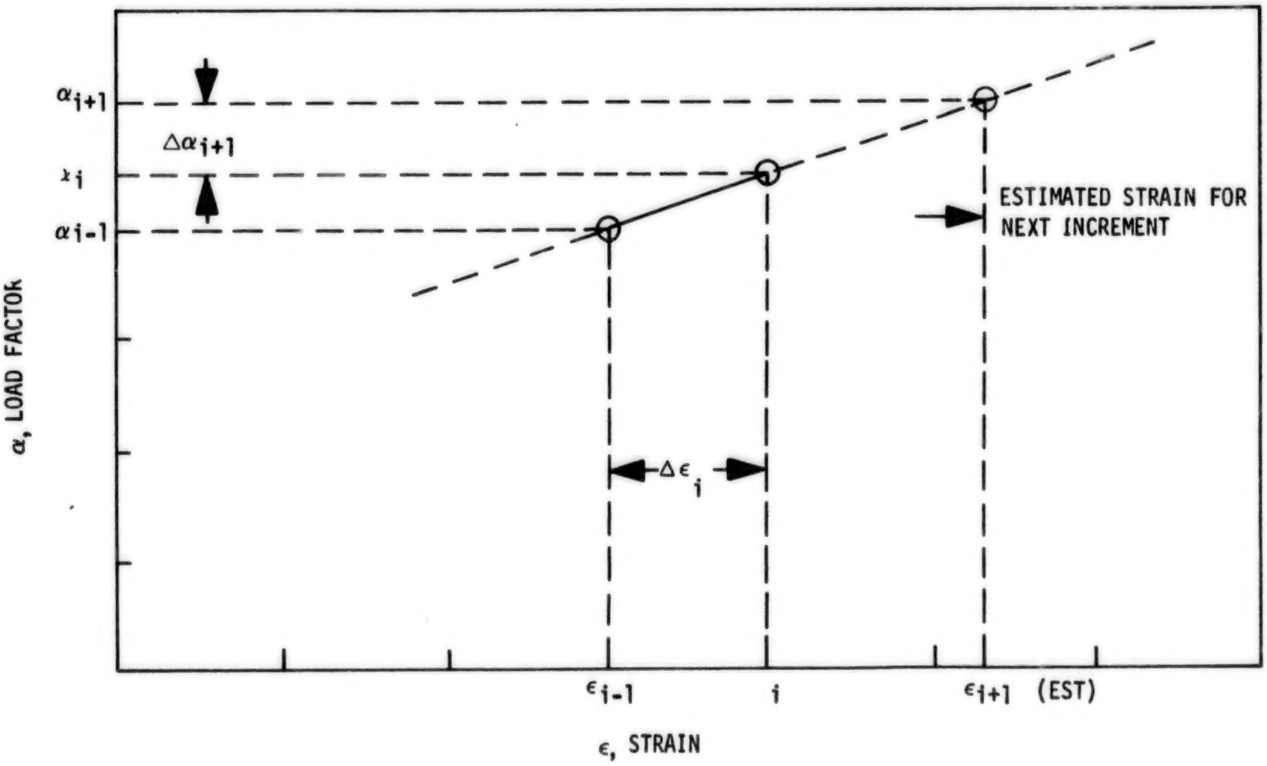


Figure 3. Method of extrapolating previous strains to produce estimate of next strain.

4. MISCELLANEOUS GENERAL PROVISIONS FOR STRUCTURAL ANALYSIS

4.1 AXISYMMETRIC STRUCTURES

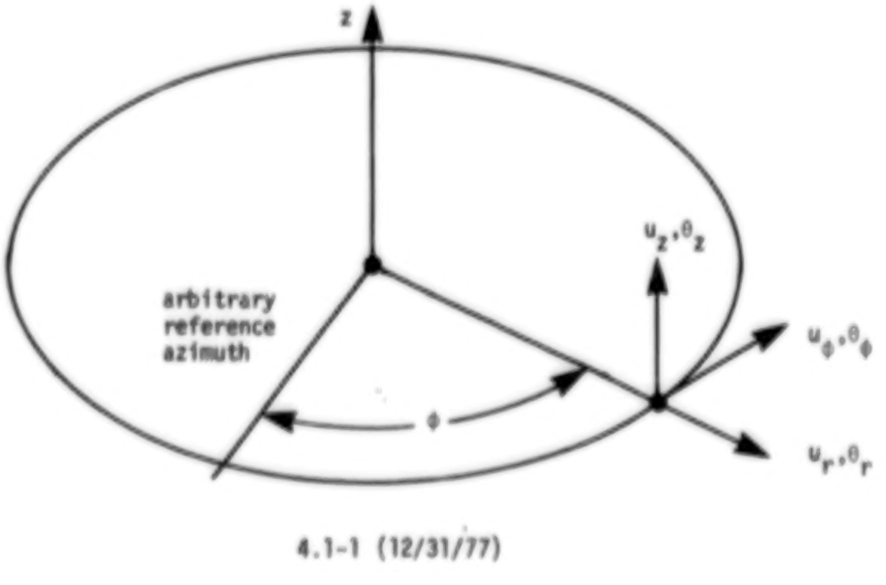
The description of finite element structural analysis presented in Chapter 3 assumes a structural model in which the degrees of freedom are defined at points in a three-dimensional space. An entirely different formulation is available in NASTRAN for analyzing axisymmetric structures. In this formulation, the degrees of freedom are the harmonic coefficients of displacement components defined on the perimeter of circles, called grid circles, which lie in planes normal to the axis of symmetry. The special features of NASTRAN's axisymmetric structural analysis capabilities are discussed below.

4.1.1 Axisymmetric Element Library

NASTRAN includes four different axisymmetric structural elements. They are the conical shell element (Section 5.9), the toroidal shell element (Section 5.10), and the triangular and trapezoidal solid ring elements (Section 5.11). The reader is referred to the sections cited for details. No attempt has been made to make these elements compatible with each other, or with "ordinary" structural elements. The only axisymmetric elements that can be used together in the same problem are the triangular and trapezoidal solid-ring elements. The conical shell element is the only element that accepts nonaxisymmetric loads. The others require that the loading be axisymmetric.

4.1.2 Coordinate Systems

The "global" coordinate system for the conical shell element, and for the solid of revolution elements, is a cylindrical coordinate system as shown below:



Force components are input parallel to the global coordinates, and displacements are output parallel to the global coordinates.

For the toroidal shell element, the "global" system is a spherical coordinate system locally tangent to the shell.

4.1.3 Harmonic Coefficients and Degrees of Freedom

The following equations are used to define harmonic coefficients:

a. Any vector component representing motions or forces in a plane that includes the z-axis:

$$v(r,\phi,z) = v_0(r,z) + \sum_{n=1}^{m} v_n(r,z)\cos(n\phi) + \sum_{n=1}^{m} v_n^*(r,z)\sin(n\phi). \tag{1}$$

b. Any vector component representing motions or forces normal to a plane that includes the z-axis:

$$u(r,\phi,z) = u_0^*(r,z) + \sum_{n=1}^{m} u_n\,(r,z)\sin(n\phi) - \sum_{n=1}^{m} u_n^*(r,z)\cos(n\phi). \tag{2}$$

The motions corresponding to different harmonic orders are uncoupled. Also, the starred and unstarred parameters are uncoupled. The degrees of freedom are the coefficients (u_{rn}, $u_{\phi n}$, u_{zn}, θ_{rn}, $\theta_{\phi n}$, θ_{zn} and their "starred" counterparts) at discrete "grid circles." Note the (-) sign preceding the starred series for $u(r,\phi,z)$. Because of the (-) sign, the starred parameters describe motions that are shifted $\frac{\pi}{2n}$ in azimuth from the motions described by the unstarred parameters, since

$$-\cos(n\phi) = \sin(n\phi - \tfrac{\pi}{2}), \tag{3}$$

and

$$\sin(n\phi) = \cos(n\phi - \tfrac{\pi}{2}). \tag{4}$$

The practical effect of the (-) sign in Equation 2 is that the stiffness matrices for the starred parameters are identical to the stiffness matrices for the unstarred parameters for n>o. Note that the unstarred coefficients represent motions that are symmetrical with respect to $\phi = 0$, and that the starred coefficients represent antisymmetrical motions.

The harmonic order, n, represents an additional dimension of the vector space that is not present in "ordinary" structural analysis. The number of degrees of freedom per grid circle is equal to 6(2m + 1).

The solutions for different unstarred harmonic orders are calculated in the same run. For statics and inertia relief problems only (Rigid Formats 1 and 2, see Section 3.2), the results for unstarred harmonic orders may be combined with the results for starred harmonic orders, thereby providing solutions for general unsymmetric loading.

For vibration mode analysis (Rigid Format 3, see Section 9.1), the user selects the highest order, m, and all modes of order m and lower are calculated in the same run. There is no provision for selecting individual harmonic orders.

The presence of harmonic coefficients complicates the selection of the order in which degrees of freedom are processed. In NASTRAN, the degrees of freedom are sequenced first by location and then by harmonic order and last by symmetry (starred or unstarred). Thus, all v_0 coefficients precede all v_1 coefficients, etc. Since no coupling between different orders is permitted, this is a sensible arrangement that minimizes bandwidth.

4.1.4 Application of Loads

The following types of static loads are available for use with the conical shell element:

a. Concentrated forces and moments applied at points on grid circles.

b. Uniform line load on a sector of a grid circle.

c. Uniform pressure load on a region bounded by two gird circles and two meridians.

d. Harmonic components of force and moment along grid circles.

e. Gravity loads. The gravity vector may be arbitrarily oriented. It operates on the global mass matrix and generates zero and first harmonic loads.

f. Thermal loads. The temperature is defined at specified points on grid circles and is linearly interpolated. The provision for harmonic components of temperature described on Page 5.9-28 has not been implemented. The temperature on grid circles is used by element routines to compute thermal loads.

g. Enforced displacements at grid circles. Harmonic components are constrained to user-specified values.

The only static loads that can be applied when the solid-ring elements are used are uniform symmetrical line forces on grid circles and thermal loads. The toroidal shell element accepts uniform line forces and line moments on the grid circles. It also accepts thermal loading.

At present, very special procedures are required to generate dynamic loads. The computer generates internal gridpoint numbers for each harmonic at each grid circle. If the user knows the algorithm by which gridpoint numbers are assigned, he can reference the internally assigned numbers and apply a load to them. The procedure is described in Section 4.6 of the Programmer's Manual.

4.1.5 Differential Stiffness

Differential stiffness (see Chapter 7) is available for the conical shell element only. It provides a linear buckling capability for symmetrically loaded shells of revolution. If a non-symmetric loading is applied, NASTRAN extracts the zero harmonic component of the load and then computes the resulting differential stiffness for all harmonics. It will also compute the buckling modes for all harmonics.

4.1.6 Hydroelastic Capability

The NASTRAN hydroelastic capability is described in Section 16.1. The properties of the fluid are assumed to be axisymmetric, and a Fourier series expansion is used. At present, the properties of the structure must be expressed with ordinary nonaxisymmetric structural elements in hydroelastic problems.

4.2 STRUCTURAL MATERIAL PROPERTIES

In NASTRAN structural material properties are normally specified with a Material Property Card. The only exceptions are the scalar damper and the scalar spring whose material properties are imbedded in their spring and damping constants. The material properties defined on Material Property cards include density, elastic moduli, thermal expansion coefficients, allowable stresses used in calculating margins of safety, and structural damping coefficients. All of the material properties can be made functions of temperature and elastic moduli can be made functions of stress for piecewise linear analysis (see Section 3.8).

At present three different types of material property cards are available. Table 1 summarizes the availability of the material property types for each of the NASTRAN structural elements. The manner in which elastic moduli are treated by each of the Material Property Cards is as follows:

MAT1 - specifies values of E, ν, and/or G for isotropic materials. When two of the three parameters are specified, the third is computed from $G = E/2(1+\nu)$. If all three parameters are specified, the value specified for G is replaced by this formula for surface and surface of revolution elements. For solid and solid of revolution elements all three parameters are used in the form:

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 & 0 & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{Bmatrix} \tag{1}$$

Note that the material is not isotropic when $G \neq E/2(1+\nu)$. For solid elements the material axes to which Equation 1 refers are the axes of the basic coordinate system. The material axes for solid of revolution elements are defined on Page 5.11-22.

MAT2 - specifies a general anisotropic stress-strain relationship in the form

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{12} & G_{22} & G_{23} \\ G_{13} & G_{23} & G_{33} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad . \tag{2}$$

This format is available for flat surface elements only, as shown in Table 1.

MAT3 - specifies a general orthotropic stress-strain relationship with respect to three perpendicular axes of symmetry in the form

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{yx}}{E_y} & -\frac{\nu_{zx}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xy}}{E_x} & \frac{1}{E_y} & -\frac{\nu_{zy}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xz}}{E_x} & -\frac{\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} \quad . \tag{3}$$

The matrix is symmetric so that

$$\frac{\nu_{xy}}{\nu_{yx}} = \frac{E_x}{E_y} \; ; \; \frac{\nu_{yz}}{\nu_{zy}} = \frac{E_y}{E_z} \; ; \; \frac{\nu_{zx}}{\nu_{xz}} = \frac{E_z}{E_x} \quad . \tag{4}$$

The inverse of the matrix in Equation 3 is of a similar form as that given in Equation 3 on Page 5.11-4. The MAT3 card is available for surface of revolution and solid of revolution elements only. These elements employ appropriate subsets of the (6x6) matrix.

The coordinate axes for the NASTRAN structural elements are defined as follows:

Linear elements (RØD, CØNRØD, and TUBE) have an element x-axis which points from end A to end B of the element. Positive extensional forces are tension; and positive torques are defined by the right-hand rule. The material properties are E (for tension) and G (for torsion).

The linear bending element (BAR), has an element x-axis which points from end A to end B of the bar. The ends may be offset from the grid points by rigid connections. The element y-axis is determined by a user specified vector $\vec{V}$. The y-axis is in the plane of the vector $\vec{V}$ and the x-axis, and it is perpendicular to the x-axis. The z-axis forms a right-hand system. The material properties are E (for extension and bending) and G (for torsion and transverse shear). For composite beams, a reference value of E can be chosen, and the user can then evaluate the effective area and moments of inertia. Similarly for G, the user can evaluate the effective torsional rigidity (J) and transverse shear factors (K_y, K_z) (see Section 5.2.1). Thus, E and G are sufficient to describe sandwich type beams.

The surface elements have an element coordinate system internal to each element. The element lies in its x-y plane, with the origin at the first listed grid point, and the second listed grid point on the x-axis. Element forces and stresses are given in this coordinate system. References can be made to different material properties for membrane, bending and transverse shear deformations to account for sandwich plates. Either MAT1 or MAT2 type materials may be used. The material matrix (if it is type 2) may be specified in a material coordinate system whose x-axis makes an angle θ with the x-axis of the element coordinate system, as shown in Figure 1.

The theoretical development in Section 5.8.2.4 allows for a 2 x 2 transverse shear matrix

$$\begin{Bmatrix} \gamma_x \\ \gamma_y \end{Bmatrix} = \begin{bmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{bmatrix} \begin{Bmatrix} V_x \\ V_y \end{Bmatrix} \quad , \tag{5}$$

relating transverse shear deflections to shear forces. At present $J_{xx} = J_{yy} = 1/Gh$ and $J_{xy} = 0$; where G is the value specified on a MAT1 card (0.0 implies G is infinite). The entire matrix is set equal to zero if a MAT2 card is used.

The solid elements use the basic coordinate system and allow only isotropic material properties, except as noted above in connection with Equation 1.

The surface of revolution elements have s (meridonal), ϕ (azimuthal), z (normal) coordinate systems in place of x, y and z. The conical shell can specify separate isotropic (MAT1) properties for membrane, bending and transverse shear. The toroidal shell (zero harmonic only, no transverse shear) has a single 2 x 2 matrix

ORIGINAL PAGE IS
OF POOR QUALITY

$$\begin{Bmatrix} \sigma_s \\ \sigma_\phi \end{Bmatrix} = \frac{1}{1-\nu_{\phi s}\nu_{s\phi}} \begin{bmatrix} E_s & E_s\nu_{\phi s} \\ E_\phi \nu_{s\phi} & E_\phi \end{bmatrix} \begin{Bmatrix} \epsilon_s \\ \epsilon_\phi \end{Bmatrix} \quad , \tag{6}$$

where s and ϕ replace x and y and where the E's and ν's may come from a MAT1 or a MAT3 format.

The solid of revolution elements use a cylindrical r, ϕ, z coordinate system. Either MAT1 or MAT3 formats can be used.

Thermal expansion coefficients are also specified on the Material Property cards. On a MAT1 card the thermal expansion is assumed isotropic. On a MAT2 card

$$\begin{Bmatrix} \epsilon_{xt} \\ \epsilon_{yt} \\ \gamma_t \end{Bmatrix} = \begin{Bmatrix} A_1 \\ A_2 \\ A_{12} \end{Bmatrix} T \quad , \tag{7}$$

On a MAT3 card

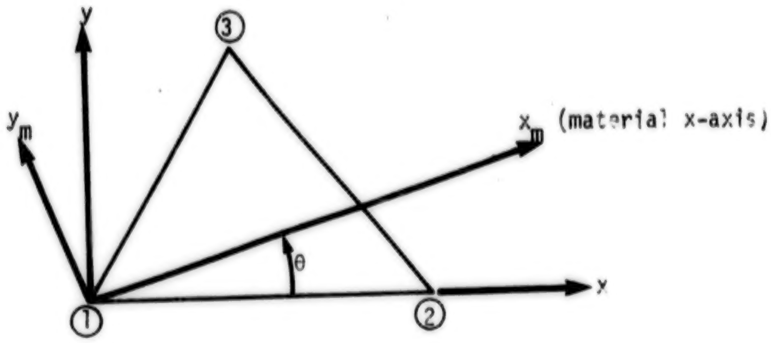
$$\begin{Bmatrix} \epsilon_{xt} \\ \epsilon_{yt} \\ \epsilon_{zt} \end{Bmatrix} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix} T \quad . \tag{8}$$

Note that the material is assumed to be symmetrical with respect to its axes on a MAT3 card.

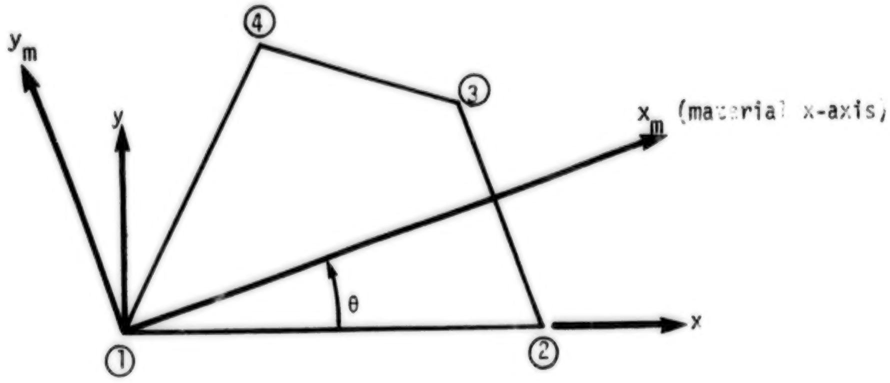
Table 1. Existing NASTRAN material capability.

Element Type	Element Name	Kind of Strain	MAT type
Linear	RØD, CØNRØD, TUBE	tension, torsion	1
	BAR	tension, torsion, bending, shear	1
Surface	TRMEM, QDMEM	membrane	1, 2
	TRPLT, QDPLT, TRBSC	bending transverse shear	1, 2^* 1, (2^*)
	TRIA1, QUAD1	membrane bending transverse shear	1, 2 1, 2^* 1, (2^*)
	SHEAR, TWIST	shear	1
Solid	TETRA, WEDGE, HEXA1-2	3-dimensional	1
Surface of Revolution	CØNEAX	membrane bending shear	1 1 1
	TØRDRG	ε_s, ε_θ	1, 3
Solid of Revolution	TRIARG, TRAPRG	ε_r, ε_z, ε_θ, γ_{rz}	1, 3

*If MAT2 is used, the shear flexibility is 0.0


y
③
y_m
x_m (material x-axis)
θ
x
①
②

(a) Triangle


y_m
④
③
y
x_m (material x-axis)
θ
x
①
②

(b) Quadrilateral

Figure 1. Material axes for surface elements.

4.3 MANUAL SINGLE-STAGE SUBSTRUCTURE ANALYSIS

Substructure partitioning, as here defined, is a procedure in which the structural model is divided into separate parts which are then processed, in separate computer executions, to the point where the data blocks required to join each part to the whole are generated. The subsequent operations of merging the data for the substructures, and of obtaining solutions for the combined problem, are performed in one or more subsequent executions, after which detailed information for each substructure is obtained by additional separate executions.

Substructure partitioning may be required for logistic reasons in problem preparation, for reasons of computational efficiency, or simply because the high-speed or peripheral storage capacity of the computer is exceeded by the data generated in the solution of the problem as a single structure. The logistic reasons refer to the possibility that the task of preparing the mathematical model of the structure may be assigned to separate groups which work at different places and times or at different rates and which require frequent access to the computer in order to check their work. It may, in such situations, be cost effective to combine the results of the separate computer runs, rather than their separate input data decks.

Sections 3.5.3 and 3.5.4 describe a matrix partitioning procedure which is available as an internal part of the rigid formats and which does not, therefore, qualify as substructure partitioning. It divides the degrees of freedom into two sets: the "a" set, u_a, which is retained, and the "o" set, u_o, which is omitted in subsequent processing. The manner in which this procedure may be used to generate true substructure partitioning is illustrated in Figure 1. If the u_a set is selected as shown, the structural matrices for the u_o grid points in different regions will be uncoupled from each other. For example, the nonzero terms in $[K_{oo}]$ will only occur in diagonal partitions as shown in Equation 1 below.

$$[K_{oo}] = \begin{bmatrix} K_{oo}^{(1)} & 0 & 0 & 0 \\ 0 & K_{oo}^{(2)} & 0 & 0 \\ 0 & 0 & K_{oo}^{(3)} & 0 \\ 0 & 0 & 0 & \text{Etc.} \end{bmatrix} . \qquad (1)$$

The generation of the reduced stiffness matrix $[K_{aa}]$ and the reduced load vector $\{P_a\}$ by Equations 28 and 29 of Section 3.5.3 can then proceed independently for each region (i.e., substructure). Specifically,

$$[K_{aa}] = [K_{aa}^{(p)}] + \sum_i [K_{aa}^{(i)}] \, , \tag{2}$$

$$\{P_a\} = [P_a^{(p)}] + \sum_i \{P_a^{(i)}\} \, , \tag{3}$$

where the reduced stiffness matrix for each substructure

$$\begin{aligned} [K_{aa}^{(i)}] &= [K_{ao}^{(i)}] \; [G_o^{(i)}] + [\overline{K}_{aa}^{(i)}] \\ &= - [K_{ao}^{(i)}] \; [K_{oo}^{(i)}]^{-1} \; [K_{ao}^{(i)}]^T + [K_{aa}^{(i)}] \, , \end{aligned} \tag{4}$$

and the reduced load vector for each substructure

$$\{P_a^{(i)}\} = \{\overline{P}_a^{(i)}\} + [G_o^{(i)}]^T \{P_o^{(i)}\} \, . \tag{5}$$

The terms $[K_{aa}^{(p)}]$ and $[P_a^{(p)}]$ in Equations 2 and 3 represent terms added by the user in a later stage.

Substructural analysis by the NASTRAN substructuring technique is logically performed in at least three phases, as follows:

Phase I: Analysis of each individual substructure by NASTRAN to produce a description, in matrix terms, of its behavior as seen at the boundary degrees of freedom, u_a.

Phase II: Combination of appropriate matrices from Phase I and the inclusion, if desired, of additional terms to form a "pseudo-structure" which is then analyzed by NASTRAN.

Phase III: Completion of the analysis of individual substructures using the $\{u_a\}$ vector produced in Phase II.

The NASTRAN substructuring technique is available for all rigid formats, except piecewise linear static analysis. In the case of static rigid format 1, no additional approximations are introduced into the calculation by the substructuring operation. In the case of dynamic rigid formats, the Guyan reduction is employed in Phase I, which restricts the dynamic degrees of freedom to the $\{u_a\}$ set (see Section 3.5.4). Thus, it is advisable, when solving a dynamic problem, to include some degrees of freedom at interior points in $\{u_a\}$. Also, $\{u_a\}$ must, of course, include all degrees of freedom that are connected to more than one substructure.

Under certain circumstances, the substructure analysis may use more than three phases. For example, if differential stiffness is included, five phases are required as follows: (1) initial static preload analysis of each substructure; (2) combination static preload analysis; (3) recovery of static preload stress data, and calculation of the differential stiffness matrix for each substructure; (4) combination analysis, including differential stiffness; and (5) completion of the analysis of individual substructures. Note that rigid format 4, Static Analysis with Differential Stiffness, is not used in the analysis sequence. A similar procedure is followed in the case of a buckling analysis, except that it is advisable to include some degrees of freedom at interior points in $\{u_a\}$; otherwise the influence of differential stiffness on the buckling mode shape at interior points will be ignored. Another example where more phases are used is an analysis where the substructures are first combined into groups, and the groups are then combined into a complete "pseudostructure."

As can be seen, a flexible substructuring capability is necessary to accommodate all practical uses. This is provided by using the ALTER feature (see Section 1.3) to modify existing rigid formats according to the user's requirements.

Figure 2 shows a typical flow diagram for the operation of substructuring in NASTRAN. It involves the application of three separate phases of NASTRAN execution to two substructures. In the NASTRAN Phase I execution, the stiffness matrix $[K_{aa}]$ and (if needed) the static load vector $\{P_a\}$ are computed independently for each substructure. In dynamic analysis, the matrices $[M_{aa}]$, $[K^4_{aa}]$, and $[B_{aa}]$ are also computed. All of these data are copied onto a user tape via the user module ØUTPUT1, which is altered into the rigid format. The computation of the dynamic load vector is delayed until Phase II.

The first step in the NASTRAN Phase II execution is to merge the reduced matrices formed in Phase I. This is done by the existing MERGE and ADD modules which are altered into the NASTRAN rigid format selected for Phase II. The MERGE operation requires knowledge of the interconnections between the degrees of freedom in the substructures. This information is contained in a partitioning matrix, each of whose columns corresponds to a particular substructure. The rules for generating the partitioning matrix are explained in Section 1.10 of the User's Manual. In Phase II, the degrees of freedom in $\{u_a\}$ constitute a set of scalar points, which is redesignated as the $\{u_g\}$ set of the "pseudo-structure" and on which all normal NASTRAN operations may be performed. In particular, direct matrix input (DMI), single and multi-point constraints (SPC's and MPC's), and both static and dynamic loads may be applied. The partitioning matrix is employed by the user to identify the degrees of freedom in $\{u_g\}$.

In Phase III, eacn NASTRAN substructure execution is restarted with the partition of the Phase II $\{u_g\}$ vector corresponding to the $\{u_a\}$ vector for each substructure. All normal data reduction procedures may then be applied. In dynamic analysis, Phase III can be omitted if output requests are restricted to the response quantities in the u_a set.

In a dynamic analysis the user may, if he wishes, employ the Guyan reduction in Phase II. The complete substructure analysis then involves a "double reduction" in which some degrees of freedom are eliminated in Phase I and some are eliminated in Phase II. This is useful because, as noted earlier, the $\{u_a\}$ vector generated in Phase I contains all of the degrees of freedom on the boundaries between substructures, as well as a selected set of freedom at interior points. The density of the boundary freedoms may well be greater than necessary, and these freedoms can be removed for the sake of economy by the second Guyan reduction. The final set of freedoms retained for dynamic analysis will be those actively selected by the user and no more. The double reduction technique is recommended for structures with very many static degrees of freedom, where it will be competitive with component mode synthesis (see Section 14.1) in many cases.

Detailed instructions for the NASTRAN substructuring procedures are given in Section 1.10 of the User's Manual.

ORIGINAL PAGE IS
OF POOR QUALITY

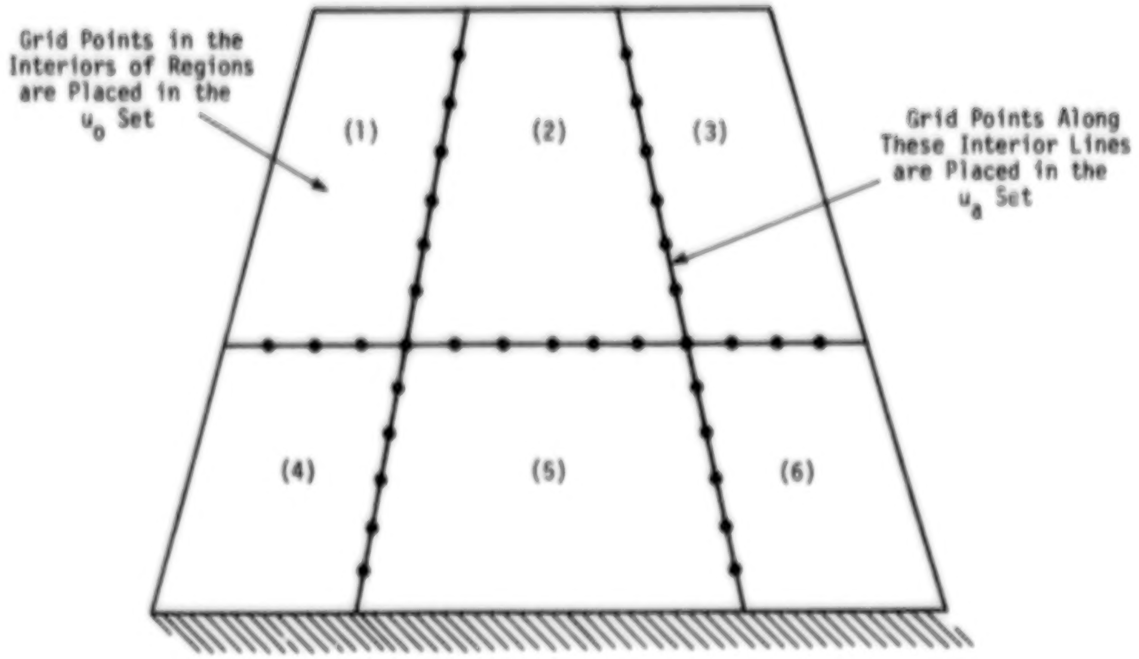
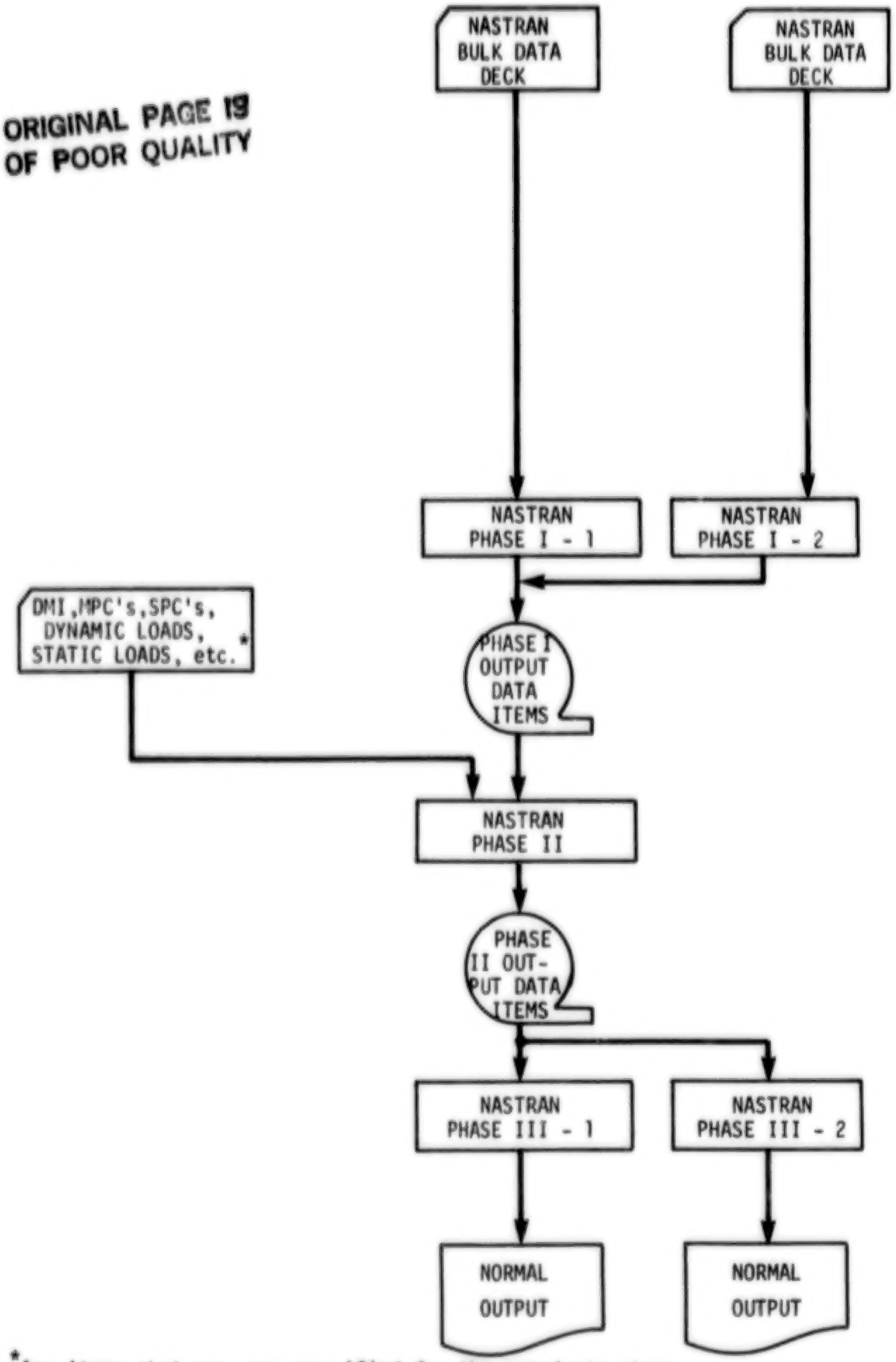

Grid Points in the Interiors of Regions are Placed in the u_o Set
Grid Points Along These Interior Lines are Placed in the u_a Set
(1)
(2)
(3)
(4)
(5)
(6)

Figure 1. Use of partitioning to decouple regions of the structure.

148


NASTRAN BULK DATA DECK
NASTRAN BULK DATA DECK
NASTRAN PHASE I - 1
NASTRAN PHASE I - 2
DMI, MPC's, SPC's, DYNAMIC LOADS, STATIC LOADS, etc.*
PHASE I OUTPUT DATA ITEMS
NASTRAN PHASE II
PHASE II OUTPUT DATA ITEMS
NASTRAN PHASE III - 1
NASTRAN PHASE III - 2
NORMAL OUTPUT
NORMAL OUTPUT

*Any items that are user-specified for the pseudostructure

Figure 2. Flow diagram for NASTRAN substructuring.

4.4 FULLY STRESSED DESIGN OPTIMIZATION

NASTRAN includes a method of design optimization for linear static analysis (Rigid Format 1) based on automation of a relatively simple strategy known as "fully stressed design." According to this concept, the cross-sectional properties of each structural element are changed at each design iteration to produce a limit stress (zero margin of safety) somewhere within the element, on the assumption that the loads carried by the element are unaffected by changes in its cross-sectional properties. The assumption is strictly true only for statically determinate structures. In indeterminate structures of low redundancy, the assumption is not badly in error, so that a few repetitions of the algorithm will produce a stress distribution throughout the structure which has very nearly a zero margin of safety in every element, i.e., a "fully stressed" design. In structures of high redundancy, the procedure will converge more slowly (if at all), and modifications of the basic strategy may be required to achieve convergence. There is, furthermore, no assurance that the fully stressed design of a highly redundant structure will be an optimum design in any meaningful sense. It is relatively easy to construct examples in which the procedure converges to a "pessimum" design. Consider, for example, the simple case of two parallel rods which are rigidly connected together at their ends and which differ only in their allowable stresses. Since in this case the stresses in the two rods are equal regardless of their areas, the algorithm will increase the area of the weaker rod at the expense of the stronger, and in the limit only the weaker rod will remain.

From this example it is seen, at the least, that a fully stressed design algorithm cannot be used uncritically. It is, nevertheless, very attractive because of its basic simplicity, and it will produce excellent designs in many practical cases. On the other hand, due to its inherent limitations, it is not deemed to be worthy of a great deal of refinement. Consequently, in the NASTRAN version, the criteria used to resize elements have been kept simple. User experiences with the method may lead to improvements in later versions.

The physical quantities involved in the design algorithm are: properties, A; stresses, σ; and stress limits, σ_ℓ. The properties may include thicknesses, cross-sectional areas or moments of inertia. Most NASTRAN elements have several independent properties. They also have several types of stresses and several places where stresses can be evaluated. The stress limits include those for tension, compression and shear. For the simple case of an element with one property, the design iteration algorithm is as follows. Let

$$\alpha = \mathrm{Max}\left|\frac{\sigma}{\sigma_\ell}\right| \quad , \tag{1}$$

where the search for a maximum value is extended over all user-identified stress components and locations, and also over all designated loading cases. The new property for the element is evaluated from the old property by the formula

$$A_{new} = A_{old}\left(\frac{\alpha}{\alpha + (1-\alpha)\gamma}\right) \quad , \tag{2}$$

where γ is a parameter selected by the user. For $\gamma = 1$ (the default value), Equation 2 becomes

$$A_{new} = \alpha\, A_{old} \quad . \tag{3}$$

If the product σA were invariant, Equation 3 would give

$$\sigma_{new} = \frac{A_{old}}{A_{new}}\sigma_{old} = \frac{1}{\alpha}\sigma_{old} \quad , \tag{4}$$

so that the value of σ_{new} would just be equal to the limit stress in this special case.

For $\gamma = 0$, it is seen that $A_{new} = A_{old}$, and for values intermediate between zero and one the property is changed by less than a factor of α. Thus γ is a parameter which moderates the property changes at each iteration and it may be employed by the NASTRAN user to improve the convergence of the algorithm.

The algorithm is modified by several other practical considerations. For example, the user may limit the range through which any property may be varied. In addition, a given property may be applied to several elements simultaneously via standard NASTRAN property cards. In this case the search indicated by Equation 1 will be extended to all elements which reference the same property card and the property will be changed uniformly for all such elements. Thus, the user can control the fineness of the property distribution by using a larger or smaller number of property cards. Finally, the optimization procedures will be applied only to those property cards which are called out by the user.

The number of iterations is controlled by a user-supplied convergence parameter, ε, and a user-supplied upper limit on the number of iterations. The algorithm is continued until either the limiting number of iterations is reached, or until the values of the α's for all properties which are not at their upper or lower limits are within ε of unity.

FULLY STRESSED DESIGN OPTIMIZATION

For elements with more than one cross-sectional property (such as the BAR element), the simple expedient has been chosen to change all of the properties according to a fixed rule. Thus, in the case of the BAR, the moments of inertia are changed in direct proportion to the change in area. This is equivalent to the assumption that each BAR has a thin-walled cross-section whose thickness is being changed uniformly. The details of the rule are given in Table 1 for each of the elements whose properties can be changed. In the table, the basic property is the one whose new value is calculated by Equation 2. It is seen that all related properties are changed in direct proportion to the basic properties except in the case of the homogeneous plate elements, TRIA2 and QUAD2.

The procedures for elements with more than one cross-sectional property are admittedly crude and they cannot be used for the detailed design of individual elements. The incorporation of more elaborate procedures has, however, been judged to be unwise for the present, due to the inherent limitations of the fully stressed design algorithm. Indeed, it is not clear that any fully automated general purpose design procedure can successfully cope with the simultaneous requirements of overall and detailed design.

The calculations are performed in two modules, ØPTPR1 and ØPTPR2 (Property Optimization Processor, Phases 1 and 2). The first module creates a table of the relevant quantities for each identified element property card, and the second module calculates the changes in the values of the properties.

The output of the analysis includes a revised set of element property cards in addition to all normal classes of output data.

ORIGINAL PAGE IS OF POOR QUALITY

Table 1. Rules for changing the properties of elements.

Element	σ, Stress Value Inspected	α, Scale Factor for Use in Equation 2	Basic Property	Related *** Properties
ROD TUBE	Axial Torsional*	$\alpha=\max\left(\frac{\sigma}{\sigma_t}, \frac{-\sigma}{\sigma_c}, \frac{\tau}{\sigma_s}\right)$	A* O.D.	J
BAR	Maximum axial tension and axial compression stresses at user-selected points in the cross sections at ends A and B.	$\alpha=\max\left(\frac{\sigma_{a1}}{\sigma_t}, \frac{\sigma_{b1}}{\sigma_t}, \frac{-\sigma_{a2}}{\sigma_c}, \frac{-\sigma_{b2}}{\sigma_c}\right)$	A**	J, I_1, I_2, I_{12}
TRMEM QDMEM	σ_{p1} = MAX PRINCIPAL σ_{p2} = MAX PRINCIPAL τ_m = MAX SHEAR	$\alpha=\max\left(\frac{\sigma_{p1}}{\sigma_t}, \frac{-\sigma_{p2}}{\sigma_c}, \frac{\tau_m}{\sigma_s}\right)$	t	--
TRPLT, QDPLT TRBSC TRIA1, QUAD1	Same as above except at outer fiber z_1, z_2.	$\alpha=\max\left(\frac{\sigma_{p1}}{\sigma_t}, \frac{-\sigma_{p2}}{\sigma_c}, \frac{\tau_m}{\sigma_s}\right)$ (for both z_1 and z_2)	I t	t (for transverse shear) I
TRIA2, QUAD2	Same as above	Same as above	t	$I = t^3/12$
SHEAR	τ_m = MAX SHEAR	$\alpha=\left(\frac{\lvert\tau_m\rvert}{\sigma_s}\right)$	t	--

* If A is zero, J will be used in its place.

** If A is zero, I1 will be used in its place.

*** The related properties are changed in direct proportion to the basic property unless otherwise indicated.

4.5 CYCLIC SYMMETRY

Many structures, including pressure vessels, rotating machines, and antennas for space communications, are made up of virtually identical segments that are symmetrically arranged with respect to an axis. There are two types of cyclic symmetry as shown in Figures 1 and 2: simple rotational symmetry, in which the segments do not have planes of reflective symmetry and the boundaries between segments may be general doubly-curved surfaces; and dihedral symmetry, in which each segment has a plane of reflective symmetry and the boundaries between segments are planar. In both cases, it is most important for reasons of economy to be able to calculate the thermal and structural response by analyzing a subregion containing as few segments as possible.

Principles of reflective symmetry (which are not, in general, satisfied by cyclicly symmetric bodies) can reduce the analysis region to one-fourth of the whole. Principles of cyclic symmetry, on the other hand, can reduce the analysis region to the smallest repeated section of the structure. Neither accuracy nor generality need be lost in the process, except that the treatment is limited to linear relationships between degrees of freedom. Special procedures for the treatment of cyclic symmetry have been added to NASTRAN. The use of cyclic symmetry allows the analyst to model only one of the identical segments. There will also be a large saving of computer time for most problems. Details of the procedures for applying cyclic symmetry are described in Section 1.12 of the User's Manual.

The term dihedral symmetry is borrowed from Herman Weyl who used it in his mathematical treatment of symmetry, Reference 1. Note that dihedral symmetry is a special case of rotational symmetry. In both cases, the body is composed of identical segments, each of which obeys the same physical laws. The distortions (deflections or temperature changes) of the segments are not independent, but must satisfy compatibility at the boundaries between segments. Cyclic transforms can be defined which are linear combinations of the distortions of the segments. The transformed equations of compatibility are such that the "transformed segments" are coupled singly or in pairs which can be solved independently. This feature results in a significant reduction of computational effort beyond the normal possibilities of substructure analysis.

4.5.1 Theory for Rotational Symmetry

The total body consists of N identical segments, which are numbered consecutively from 1 to N. The user supplies a NASTRAN model for one segment. All other segments and their coordinate systems are rotated to equally-spaced positions about the polar axis. The boundaries must be conformable; i.e., when the segments are put together, the grid points and the displacement coordinate systems of adjacent segments must coincide; thus no point may be on the axis. This is easiest to insure if a cylindrical or spherical coordinate system is used, but such is not required. The user will also supply a paired list of grid points on the two boundaries of the segment where connections will be made. For static analysis the user may also supply a set of loads and/or enforced displacements for each of the N segments.

The two boundaries will be called sides 1 and 2. Side 2 of segment n is connected to side 1 of segment n+1, see Figure 1. Thus, the components of displacement satisfy

$$u_1^{n+1} = u_2^n \qquad n = 1 \ldots N\ , \tag{1}$$

where the superscript refers to the segment index and the subscript refers to the side index. This applies to all degrees of freedom which are joined together. Also let $u_2^{N+1} = u_1^1$, so that Equation 1 will refer to all boundaries. Equation 1 is the equation of constraint between the physical segments.

The rotational transformation is given by

$$u^n = \bar{u}^o + \sum_{k=1}^{k_L} [\bar{u}^{kc} \cos(n-1)ka + \bar{u}^{ks} \sin(n-1)ka] + (-1)^{(N/2)-1} u^{-N/2}\ , \tag{2}$$

$$a = 2\pi/N, \qquad n = 1, 2, \ldots, N,$$

where u^n can be any component of a displacement, force, stress, temperature, etc., in the n^{th} segment. The last term exists only when N is even. The summation limit $k_L = (N-1)/2$ if N is odd and $(N-2)/2$ if N is even. The transformed quantities, $\bar{u}^o$, $\bar{u}^{kc}$, $\bar{u}^{ks}$, and $\bar{u}^{N/2}$ will be referred to as symmetrical components. They are given this designation by virtue of their similarity to the symmetrical components used by electrical engineers in their analysis of polyphase networks, Reference 2. Note also the similarity of Equation 2 to a Fourier series decomposition, except that the number of terms is finite. On this account, Equation 2 could be called a finite Fourier transformation, Reference 3.

Equation 2, in matrix form, is

$$\lfloor u \rfloor = \lfloor \bar{u} \rfloor [T] \quad , \tag{3}$$

where $\lfloor u \rfloor = \lfloor u^1, u^2, u^3, \ldots, u^N \rfloor$,

and $\lfloor \bar{u} \rfloor = \lfloor \bar{u}^o, \bar{u}^{1c}, \bar{u}^{1s}, \bar{u}^{2c}, \bar{u}^{2s}, \ldots, \bar{u}^{N/2} \rfloor$.

Each element in the first row vector can represent all of the unknowns in one segment.

The expanded form of the transformation matrix is

$$[T] = \left[\begin{array}{ccccc} 1 & 1 & 1 & \cdots & 1 \\ \hline 1 & \cos a & \cos 2a & \cdots & \cos(N-1)a \\ 0 & \sin a & \sin 2a & \cdots & \sin(N-1)a \\ \hline 1 & \cos 2a & \cos 4a & & \cos(N-1)2a \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & \sin k_L a & \sin 2k_L a & \cdots & \sin(N-1)k_L a \\ \hline 1 & -1 & 1 & \cdots & -1 \end{array} \right] . \tag{4}$$

The last row exists only for even N. The transformation matrix, [T], has the property

$$[T][T]^T = [D] = \begin{bmatrix} N & & & & & & \\ & N/2 & & & & & \\ & & N/2 & & & & \\ & & & N/2 & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ & & & & & & \cdot \\ & & & & & & & N \end{bmatrix} , \tag{5}$$

i.e., the rows of T are orthogonal.

Since D is nonsingular,

$$[T][T]^T[D]^{-1} = [I] \quad . \tag{6}$$

Thus, $[T]^{-1} = [T]^T[D]^{-1}$ and

$$\lfloor \bar{u} \rfloor = \lfloor u \rfloor [T]^{-1} = \lfloor u \rfloor [T^T D^{-1}] \quad . \tag{7}$$

In summation form, Equation 7 becomes

$$\left.\begin{aligned}
\bar{u}^o &= (1/N) \sum_{n=1}^{N} u^n \quad , \\
\bar{u}^{kc} &= (2/N) \sum_{n=1}^{N} u^n \cos(n-1)ka \quad , \\
\bar{u}^{ks} &= (2/N) \sum_{n=1}^{N} u^n \sin(n-1)ka \quad , \\
\bar{u}^{N/2} &= (1/N) \sum_{n=1}^{N} (-1)^{n-1} u^n \text{ (N even only)} \quad .
\end{aligned}\right\} \tag{8}$$

It should be noted that Equations 8 apply to applied loads, and to internal forces, as well as to displacement components. The validity of the symmetrical components $\lfloor \bar{u} \rfloor$ to represent the motions of the system follows from the existence of $[T]^{-1}$. It remains only to show that they are useful.

The equations of motion at points interior to the segments are linear in displacements, forces, and temperatures; they are identical for all segments; and they are not coupled between segments. Thus, the equations of motion (for example, $[K]\{u\}^n = \{P\}^n$ in static analysis) can be additively combined using one of the sets of coefficients in Equations 8, thereby obtaining the equations of motion for one of the transformed variables which will have identically the same form (e.g. $[K]\{\bar{u}\}^{kc} = \{\bar{P}\}^{kc}$) as the equations of motion for one of the physical segments.

The equations of motion at points on the boundaries between segments are treated by employing the notion of a rigid constraint connecting adjacent points. To transform the compatibility equation of constraint (1), notice that

$$u_1^{n+1} = \bar{u}_1^o + \sum_{k=1}^{k_L} [\bar{u}_1^{kc} \cos nka + \bar{u}_1^{ks} \sin nka] + (-1)^n \bar{u}_1^{N/2} \quad . \tag{9}$$

157

By use of the identities cos nka = cos(n-1)ka·cos ka - sin(n-1)ka sin·ka and sin nka = sin(n-1)ka·cos ka + cos(n-1)ka·sin ka, Equation 9 may be written

$$u_1^{n+1} = \bar{u}_1^o + \sum_{k=1}^{k_L} \left[\begin{array}{l} (\bar{u}_1^{kc} \cos ka + \bar{u}_1^{ks} \sin ka)\cos(n-1)ka \\ + (-\bar{u}_1^{kc} \sin ka + \bar{u}_1^{ks} \cos ka)\sin(n-1)ka \end{array} \right] - (-1)^{n-1}\, \bar{u}_1^{N/2} \quad . \qquad (10)$$

If Equation 10 is compared to Equation 2, evaluated at side 2 as required by Equation 1, and the coefficients of terms with the same dependence on n are equated, the following equations are obtained:

$$\left. \begin{array}{c} \bar{u}_1^o = \bar{u}_2^o \quad , \\ \left. \begin{array}{r} \bar{u}_1^{kc} \cos ka + \bar{u}_1^{ks} \sin ka = \bar{u}_2^{kc} \\ -\bar{u}_1^{kc} \sin ka + \bar{u}_1^{ks} \cos ka = \bar{u}_2^{ks} \end{array} \right\} k = 1,\ldots,k_L \quad , \\ -\bar{u}_1^{N/2} = \bar{u}_2^{N/2} \quad . \end{array} \right\} \qquad (11)$$

Equations 11 are the equations of constraint for the symmetrical components. The only symmetrical components coupled by the compatibility constraints are 1c and 1s, 2c and 2s, etc. Thus, there are several uncoupled models: the K=0 model contains the $\bar{u}^o$ degrees of freedom; the K=1 model contains the $\bar{u}^{1c}$ and $\bar{u}^{1s}$ degrees of freedom, etc.

There is a somewhat arbitrary choice regarding where to transform the variables in the NASTRAN analysis. NASTRAN structural analysis can start with a structure defined with single and multipoint constraints, applied loads, thermal fields, etc., and reduce the problem to the "analysis set," $\{u_a\}$, where

$$[K_{aa}]\{u_a\} = \{P_a\} \quad . \qquad (12)$$

The vector $\{u_a\}$ contains only independent degrees of freedom. The decision was made in developing the cyclic symmetry capability to first reduce each segment individually to the "analysis" degrees of freedom, and then to transform the remaining freedoms to symmetrical components. This approach has several advantages, including elimination of the requirement to transform temperature vectors and single-point enforced displacements, because these quantities are first converted into equivalent loads. More importantly, if the "ØMIT" feature is used to remove internal degrees of freedom, it need only be applied to one segment. The ØMIT feature greatly reduces the number of degrees of freedom which must be transformed. The user specifies all constraints internal to the segments with standard NASTRAN data cards. If constraints (MPC, SPC, and/or ØMIT) are applied to degrees of freedom on the boundaries, they will take precedence over the intersegment compatibility constraints; i.e., an intersegment compatibility constraint will not be applied to any degree of freedom which is constrained in some other way. SUPØRT data cards are forbidden because they are intended to apply to overall rigid body motions and will not, therefore, be applied to each segment. In the case of static analysis, the analysis equations for the segments are

$$[K]\{u\}^n = \{P\}^n \qquad n = 1, 2, \ldots, N \quad . \tag{13}$$

The analysis equations for the symmetrical components, prior to applying the intersegment constraints, are

$$[K]\{\bar{u}\}^x = \{\bar{P}\}^x \qquad x = 0, 1c, 1s, 2c, \ldots, N/2 \quad , \tag{14}$$

where $\{\bar{P}\}^x$ is calculated using Equations 8. The matrix [K] is the same for Equations 13 and 14, and is the KAA stiffness matrix of NASTRAN for one segment.

Now consider the matter of applying the intersegment compatibility constraints. It is recognized that not all of the degrees of freedom in any transformed model can be independent, but it is easy to choose an independent set. In the independent set, $\{\bar{u}\}^K$, include all points in the interior and on boundary 1 (for both $\bar{u}^{kc}$ and $\bar{u}^{ks}$, if they exist). The values of displacement components at points on boundary 2 can then be determined from Equations 11. The transformation to the new set of independent degrees of freedom is indicated by

$$\begin{aligned} \{\bar{u}\}^{kc} &= [G_{ck}]\{\bar{u}\}^K \quad , \\ \{\bar{u}\}^{ks} &= [G_{sk}]\{\bar{u}\}^K \quad , \end{aligned} \tag{15}$$

where each row of $[G_{ck}]$ or $[G_{sk}]$ contains only a single nonzero term if it is an interior or side 1 degree of freedom and either one or two nonzero terms if it is a degree of freedom on side 2. In arranging the order of terms in $\{\bar{u}\}^K$, the user can specify either that they be sequenced with all $\{\bar{u}\}^{kc}$ terms preceding all $\{\bar{u}\}^{ks}$ terms, or that they be sequenced with $\{\bar{u}\}^{kc}$ and $\{\bar{u}\}^{ks}$ grid points alternating. It should be emphasized that the kind of vectors used in transformation of Equations 3 and 15 are quite different. In Equation 3, there is one component (or column) for each segment; in Equations 15, there is one component (or row) for each degree of freedom in a segment.

Equations 15 are used to transform Equation 14 to the following set of equations which satisfy the intersegment compatibility conditions:

$$[\bar{K}]^K\{\bar{u}\}^K = \{\bar{P}\}^K \quad , \tag{16}$$

where

$$[\bar{K}]^k = [G_{ck}^T \; K \; G_{ck} + G_{sk}^T \; K \; G_{sk}] \quad , \tag{17}$$

and

$$\{\bar{P}\}^K = [G_{ck}^T]\{\bar{P}\}^{kc} + [G_{sk}^T]\{\bar{P}\}^{ks} \quad . \tag{18}$$

Because NASTRAN has efficient sparse matrix routines, the time for the calculations indicated in Equations 17 and 18 will not be appreciable. After solving Equation 16 by decomposition and substitution, the symmetrical component variables, $\{\bar{u}\}^{kc}$ and $\{\bar{u}\}^{ks}$, are found from Equations 15. The physical segment variables, $\{u\}^n$, are found from Equation 2. The $\{u\}^n$ are NASTRAN vectors of the analysis set. They may be expanded to $\{u_g\}$ size by recovering dependent quantities. Stresses in the physical segments are then obtained via the normal stress reduction procedures.

The user may take an alternate route if he knows the transformed values, $\{\bar{P}^{kc}\}$ and $\{\bar{P}^{ks}\}$, for the forcing functions (loads, enforced displacements, and temperatures). This will, for example, be the case in a stress analysis which follows a temperature analysis of the same structural model. These data may be input directly to NASTRAN, which will convert them to the transformed load vectors, $\{\bar{P}\}^K$. Data reduction may also be performed on the transformed quantities to obtain the symmetrical components of stresses, etc.

A shortened approximate method for static analysis is available merely by setting

$$\{\bar{u}\}^K = 0 \quad , \tag{19}$$

for all K > KMAX, where KMAX is a parameter which may be set by the user. This is similar to truncating a Fourier series. The stiffness associated with larger K's (short azimuthal wave lengths) tends to be large, so that these components of displacement tend to be small.

The cyclic symmetry method can also be used in vibration analysis. The equation of motion in terms of independent degrees of freedom is

160

$$[\bar{K}^K - \omega^2\bar{M}^K]\{\bar{u}\}^K = 0 \quad , \qquad (20)$$

where $[\bar{M}]^K$ is derived by replacing [M] for [K] in Equation 17. The symmetrical components are recovered with Equation 15. No provision has been made to recover physical segment data in vibration analysis, because the physical interpretation of Equation 4 is straightforward. (Each row of [T] is a vector of the factors of the segments for one Ks or Kc index). The available output data does, however, include the symmetrical components of dependent displacements, internal forces and stresses.

4.5.2 Theory for Dihedral Symmetry

Dihedral symmetry refers to the case when each individual segment has a plane of reflective symmetry, see Figure 2. The segments are divided about their midplanes to obtain 2N half-segments, The midplane of a segment is designated as side 2. The other boundary, which must also be planar, is called side 1. The two halves of the segment are called the right "R" and left "L" halves. The user prepares model information for one R half segment. He must also supply a list of points on side 1 and another list of points on side 2.

For the case of dihedral symmetry, the cyclic transformation described earlier is used in conjunction with reflective symmetry of the segments. The two transformations are commutable, so they may be done in either order. The reflective transform for a segment is

$$\left.\begin{aligned} u^{n,R} &= u^{n,S} + u^{n,A} \quad , \\ u^{n,L} &= u^{n,S} - u^{n,A} \quad . \end{aligned}\right\} \qquad (21)$$

Here, the superscript n refers to the nth segment, and R, L the right and left halves. The superscripts S and A refer to the symmetric and antisymmetric reflective components.

In the R half segment, displacement components are referred to a right hand coordinate system; in the L half segment, displacement components are referred to a left hand coordinate system. The inverse reflective transform is

$$\left.\begin{aligned} u^{n,S} &= \frac{1}{2}\,(u^{n,R} + u^{n,L}) \quad , \\ u^{n,A} &= \frac{1}{2}\,(u^{n,R} - u^{n,L}) \quad . \end{aligned}\right\} \qquad (22)$$

Reflective symmetry is seen to be very simple. The equations of motion at interior points of the S and A half segment models are identical in form provided that unsymmetrical effects, such as Coriolis forces, are excluded.

The $u^{n,S}$ and $u^{n,A}$ components may be transformed as follows using rotational principles.

$$u^{n,x} = \bar{u}^{o,x} + \sum_{k=1}^{k_L} [\bar{u}^{kc,x} \cos(n-1)ka + \bar{u}^{ks,x} \sin(n-1)ka] + (-1)^{n-1} \bar{u}^{N/2,x} \quad , \tag{23}$$

where x may be either S (symmetric) or A (antisymmetric). The inverse transformation can be found by Equations 8 for both the symmetric and antisymmetric parts.

The constraints between the half-segments are summarized in Table 1. The constraints shown apply between points joined together at the boundary planes. "Even components" include displacements parallel to the radial planes between segment halves, rotations about the axes normal to the planes, and temperatures in a thermal analysis. "Scalar points" in a structural analysis have arbitrarily been categorized as even components. "Odd components" include displacements normal to the radial planes and rotations about axes parallel to the planes. In Table 1 the constraint equations for the S and A half-segment model are obtained by substituting Equations 21 into the equations for the L and R half-segment model. The constraint equations for the dihedral transform model are obtained by substituting for $u^{n,x}$ and $u^{n+1,x}$ from Equation 23 and comparing terms with the same dependence on n. It can be seen in the table that the k = 0 and k = N/2 models are completely uncoupled. There is coupling between the kc,S and ks,A models and also between kc,A and ks,S models. These two sets of constraint equations are related and one can be found from the other by substituting $\bar{u}^{kc,S}$ for $\bar{u}^{ks,S}$ and $\bar{u}^{ks,A}$ for $-\bar{u}^{kc,A}$ in the constraint equations. If these substitutions are made and it is noted that the equations of motion are identical at interior points, then only one coupled pair of symmetric and antisymmetric half segments need to be analyzed with different load sets for the $(\bar{u}^{kc,S}, \bar{u}^{ks,A})$ case and the $(\bar{u}^{ks,S}, -\bar{u}^{kc,A})$ case.

As in the case of general rotational symmetry, a combined set of independent degrees of freedom is formed from the half models. The independent set $\{\bar{u}\}^k$ includes all interior points, the points on side 2 of each half segment which are not constrained to zero, and new degrees of freedom, $\{\bar{u}_1\}^K$, on side 1 such that for even components in the $(\bar{u}^{kc,S}, \bar{u}^{ks,A})$ case:

$$\left.\begin{aligned} \bar{u}_1^{kc,S} &= \cos \frac{k\pi}{N} \bar{u}_1^K \quad , \\ \bar{u}_1^{ks,A} &= \sin \frac{k\pi}{N} \bar{u}_1^K \quad ; \end{aligned}\right\} \tag{24}$$

while for odd components:

$$\left.\begin{aligned} \bar{u}_1^{kc,S} &= -\sin\frac{k\pi}{N}\,\bar{u}_1^K \quad , \\ \bar{u}_1^{ks,A} &= \cos\frac{k\pi}{N}\,\bar{u}_1^K \quad . \end{aligned}\right\} \qquad (25)$$

Equations 24 and 25 are equivalent to the constraints in the third column of Table 1. The transformation to the new set of independent freedoms may be expressed as

$$\left.\begin{aligned} \{\bar{u}\}^{kc,S} &= [G_{SK}]\{\bar{u}\}^K \quad , \\ \{\bar{u}\}^{ks,A} &= [G_{AK}]\{\bar{u}\}^K \quad , \end{aligned}\right\} \qquad (26)$$

where each row of $[G_{SK}]$ or $[G_{AK}]$ contains at most a single nonzero term. The transformation matrices for the ($\bar{u}^{ks,S}$, $-\bar{u}^{kc,A}$) case are identical.

The final equation which is solved in static analysis is

$$[\bar{K}]^K\{\bar{u}\}^K = \{\bar{P}\}^K \quad , \qquad (27)$$

where the stiffness matrix

$$[\bar{K}]^K = [G_{SK}^T\,K\,G_{SK} + G_{AK}^T\,K\,G_{AK}] \quad , \qquad (28)$$

and the load vector is obtained by successive application of the inverse reflective symmetry transform, Equations 22, the inverse cyclic symmetry transform, Equations 8, and the final reduction to independent freedoms.

The form of the latter is, for the ($\bar{u}^{kc,S}$, $\bar{u}^{ks,A}$) case,

$$\{\bar{P}\}^K = [G_{SK}]^T\{\bar{P}\}^{kc,S} + [G_{AK}]^T\{\bar{P}\}^{ks,A} \quad , \qquad (29)$$

and for the ($\bar{u}^{ks,S}$, $-\bar{u}^{kc,A}$) case,

$$\{\bar{P}\}^K = [G_{SK}]^T\{\bar{P}\}^{ks,A} - [G_{AK}]^T\{\bar{P}\}^{kc,A} \quad . \qquad (30)$$

The data reduction which follows the solution of Equation 27 in static analysis includes the application of the symmetry transformation to obtain $u^{n,R}$ and $u^{n,L}$, followed by the expansion to $\{u_g\}$ size for each half-segment and the calculation of internal loads and stresses. Similar to the case of rotational symmetry, the data reduction for vibration analysis is limited to the recovery of eigenvectors, internal forces, and stresses for symmetrical component sets $\bar{u}^{kc,S}$ and $\bar{u}^{ks,A}$.

4.5.3 Advantages

The NASTRAN cyclic symmetry capability will result in a large saving of user effort and computer time for most applications. The savings result from the following effects:

1. Grid point geometry and element data are prepared for only one segment in the case of rotational symmetry or one half-segment in the case of dihedral symmetry.
2. The transformed equations are uncoupled, except within a given harmonic index, K, which reduces the order of the equations which must be solved simultaneously to 1/N or 2/N (where N is the number of segments or symmetrical half-segments) times the order of the original system.
3. Solutions may be restricted to a smaller range of the harmonic index, K, (e.g., limited to the lower harmonic orders) which results in a proportionate reduction in solution time. Some accuracy is thereby lost in the case of static analysis but not in vibration analysis.
4. In the case of static analysis, the ØMIT feature may be used to remove all degrees of freedom at internal grid points without any loss of accuracy. Since this reduction is applied to a single segment prior to the symmetry transformations, it can greatly reduce the amount of subsequent calculation.

It is instructive to compare the advantages of the NASTRAN cyclic symmetry capability with those offered by reflective symmetry and by conventional substructuring techniques. The savings offered by cyclic symmetry will always equal or exceed those provided by reflective symmetry except for possible differences due to time spent in transforming variables. For example, when an object has two planes of symmetry and two symmetrical segments (the minimum possible number in this case), the minimum model sizes are both equal to one half-segment for the two methods. They are also equal when the object has four symmetrical segments. The advantages of cyclic symmetry for these cases are restricted to those offered by the ØMIT feature in static analysis and by a higher degree of input and output data organization. Any larger number of symmetrical segments increases the advantage of cyclic symmetry because the size of the fundamental region is smaller.

A method of conventional substructuring which recognizes identical substructures can also restrict the amount of grid point geometry and element data preparation to a single substructure and can use the ØMIT feature in the same way as cyclic symmetry. The advantage which cyclic symmetry retains over conventional substructuring lies in its decomposition of degrees of freedom into uncoupled harmonic sets. This is an important advantage for eigenvalue extraction, but the advantage for static analysis is relatively small and depends in a complex manner on the number of segments and on the method of matrix decomposition.

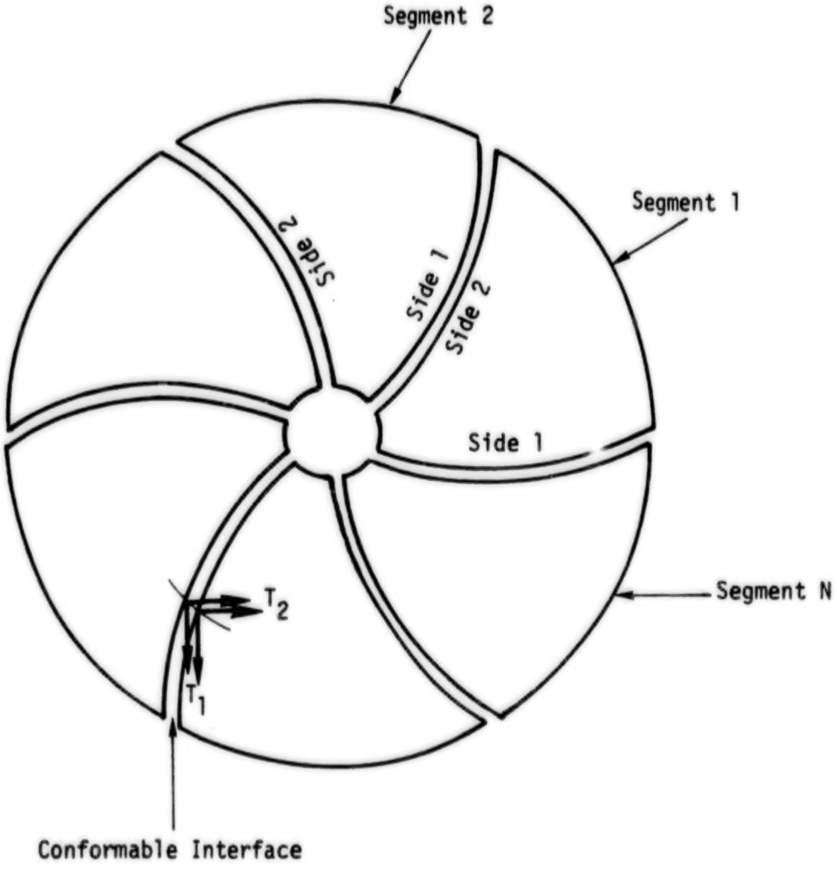
In addition to the analysis of structures made up of a finite number of identical substructures, cyclic symmetry can also be used for purely axisymmetric structures. In this case the circumferential size of the analysis region is arbitrarily selected to be some small angle, for example, one degree. Grid points are then placed on the boundary surfaces but not in the interior of the region, and the region is filled with ordinary three-dimensional elements. The principal advantage of this procedure is that ordinary three-dimensional elements are used in place of specialized axisymmetric elements. In NASTRAN the number of available types and features for ordinary three-dimensional elements far exceeds those available for axisymmetric elements, so that cyclic symmetry immediately enlarges the analysis possibilities for axisymmetric structures. In particular, the rotational symmetry option can accommodate axisymmetric structures with nonorthotropic material properties, which the available axisymmetric procedures cannot.

REFERENCES

1. Weyl, H., "Symmetry," Lectures given at the Institute for Advanced Study, reprinted in The World of Mathematics, Simon and Schuster, New York, Vol. 1, 1956, pp. 671-724.

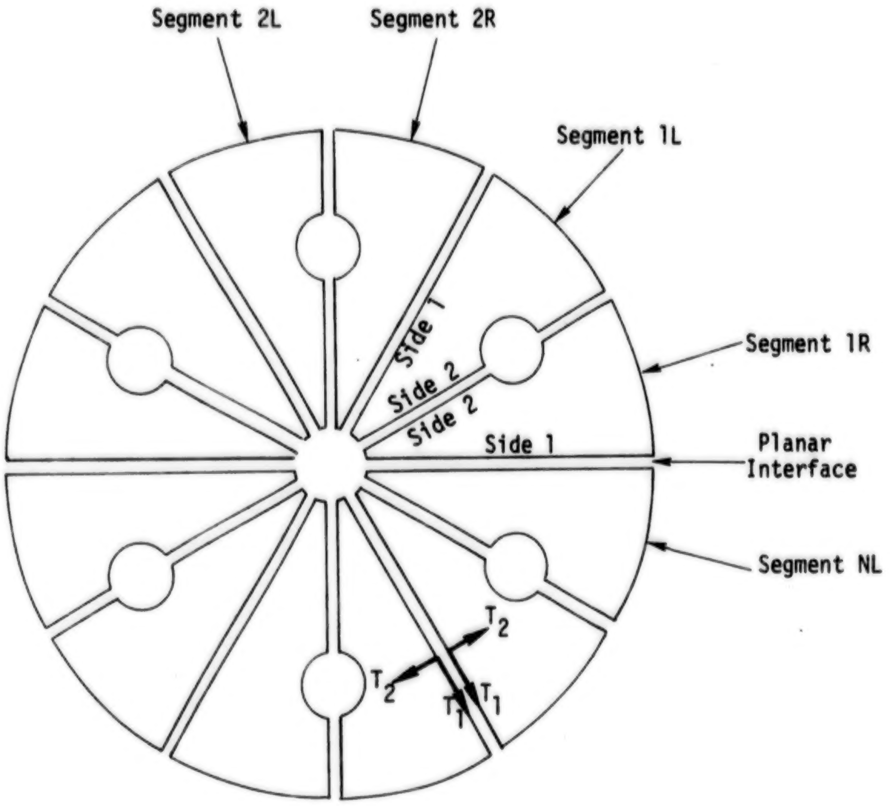
2. Fortescue, C. E., "Method of Symmetrical Coordinates Applied to the Solution of Polyphase Networks," AIEE Transactions, Vol. 37, Part II, 1918, pp. 1027-1140.

3. Hamming, R. W., Numerical Methods for Scientists and Engineers, McGraw-Hill Book Company, 1962, pp. 67-78.


Segment 2
Segment 1
Side 2
Side 1
Side 2
Side 1
Segment N
T_2
T_1
Conformable Interface

1. The user models one segment.
2. Each segment has its own coordinate system which rotates with the segment.
3. Segment boundaries may be curved surfaces. The local displacement coordinate systems must conform at the joining points. The user gives a paired list of points on Side 1 and Side 2 which are to be joined.

Figure 1. Rotational symmetry


Segment 2L
Segment 2R
Segment 1L
Side 1
Side 2
Side 2
Side 1
Segment 1R
Planar Interface
Segment NL
T_2
T_2
T_1
T_1

1. The user models one-half segment (an R segment). The L half segments are mirror images of the R half segments.
2. Each half segment has its own coordinate system which rotates with the segment. The L half segments use left hand coordinate systems.
3. Segment boundaries must be planar. Local displacement systems axes, associated with inter-segment boundaries, must be in the plane or normal to the plane. The user lists the points on Side 1 and Side 2 which are to be joined.

Figure 2. Dihedral symmetry

Table 1. Boundary Compatibility Constraints For The Case Of Dihedral Symmetry.

	L and R Half Segment Model	S and A Half Segment Model	Dihedral Transform Model
Side 1 Even Comp	$u^{n+1,R} = u^{n,L}$	$u^{n+1,S} + u^{n+1,A} = u^{n,S} - u^{n,A}$	$u^{o,A} = -su^{kc,S} + cu^{ks,A} =$ $-su^{ks,S} - cu^{kc,A} = u^{N/2,S} = 0$
Side 1 Odd Comp	$u^{n+1,R} = -u^{n,L}$	$u^{n+1,S} + u^{n+1,A} = -u^{n,S} + u^{n,A}$	$u^{o,S} = cu^{kc,S} + su^{ks,A} =$ $cu^{ks,S} - su^{kc,A} = u^{N/2,A} = 0$
Side 2 Even Comp	$u^{n,L} = u^{n,R}$	$u^{n,A} = 0$	$\bar{u}^{o,A} = \bar{u}^{kc,A} = \bar{u}^{ks,A} = \bar{u}^{N/2,A} = 0$
Side 2 Odd Comp	$u^{n,L} = -u^{n,R}$	$u^{n,S} = 0$	$\bar{u}^{o,S} = \bar{u}^{kc,S} = \bar{u}^{ks,S} = \bar{u}^{N/2,S} = 0$

where $c = \cos k\pi/N$, $s = \sin k\pi/N$

4.6 AUTOMATED MULTI-STAGE SUBSTRUCTURE ANALYSIS

The automated multi-stage substructure analysis capability allows the user to repeatedly combine and reduce structures, which in themselves may be composed of several component substructures. The resulting composite structure, or pseudostructure, may be used for performing linear static or dynamic analyses. Both static (Guyan) reduction and modal reduction (modal synthesis) may be used in creating new pseudostructures. Although some of these operations could be performed with the manual single-stage approach, described in Section 4.3, the task of controlling the sequence of operations has been automated using simple Substructure Case Control commands, and the necessary bulk data has been simplified for the user (See the User's Manual, Section 2.7). Additional capabilities, not available with the manual single-stage approach, are included. These allow geometric and symmetric transformations of entire substructures, modal synthesis of substructures, automatic identification of connected grid points based on geometry, manual identification of connectivities by grid point and component, release of selected components and multipoint constraints for special modeling conditions, independent grid point numbering of each basic substructure, and automatic internal renumbering of retained and boundary grid point degrees of freedom for the connected substructure configurations. The automated substructuring system also features an expandable substructuring data base file on which all substructuring data are automatically stored or from which data are accessed. This file is described in Section 1.10.2 of the User's Manual.

Automated multi-stage substructuring analysis is organized into three basic steps or phases. The organization of these steps is similar to the manual single-stage substructuring methods described in Section 4.3. The Phase 1 operation consists of a standard NASTRAN formulation of a basic substructure from the finite element model and its applied loads. Phase 2 performs most of the specialized operations to combine and reduce substructures, apply constraints and loads, obtain solutions, and recover data related to the basic substructures. In Phase 3, the solution vectors are used to calculate the final output for each basic substructure.

The following discussion is involved primarily with the mathematical operations performed by the automated multi-stage substructuring capabilities in NASTRAN. The specific user options and program operations are described in the User's and Programmer's Manuals, respectively.

4.6.1 Static Matrix Reduction

A powerful tool in the analysis of structures with the finite element displacement method is the matrix reduction procedure. The desired effect of this operation is to reduce the total number of degrees of freedom required to define a structure and thereby decrease the dimensions of the associated matrices and vectors. Both a Guyan static reduction and a modal synthesis reduction are available. The static reduction is described in this section, and the modal reduction is described in Section 4.6.2.

In substructure analysis, the only necessary degrees of freedom which must be retained for static analyses are those associated with the grid points which will be connected to adjacent substructures, those which will be constrained and/or those which will have loads applied to them in a subsequent analysis phase. In order to adequately represent the inertia effects of each substructure in a dynamic analysis, it will be necessary to also retain selected degrees of freedom on interior points and/or use modal reduction.

In this development, let substructure A be the structure to be reduced, and substructure B be the resultant reduced structure. The displacement vector partition $\{u_b^A\}$ defines the boundary degrees of freedom of substructure A to be retained. The vector partition $\{u_i^A\}$ contains the "interior" degrees of freedom of substructure A to be reduced.

The stiffness matrix for substructure A is partitioned so that

$$[K^A] = \left[\begin{array}{c|c} K_{bb} & K_{bi} \\ \hline K_{ib} & K_{ii} \end{array}\right] , \tag{1}$$

where the full displacement vector $\{u^A\}$ is $\left\{\begin{array}{c} u_b^A \\ \hline u_i^A \end{array}\right\}$.

Using the equations for the Guyan reduction presented in Section 3, the transformation expression is defined as

$$\left\{\begin{array}{c} u_b^A \\ \hline u_i^A \end{array}\right\} = \left[\begin{array}{c} I \\ \hline G_{ib} \end{array}\right] \left\{u^B\right\} , \tag{2}$$

or

$$\{u^A\} = [G^{AB}]\{u^B\} , \tag{3}$$

where

$$[G_{ib}] = - [K_{ii}]^{-1}[K_{ib}] . \tag{4}$$

The equations of motion for substructure A,

$$[M^A]\{\ddot{u}^A\} + [B^A]\{\dot{u}^A\} + [K^A]\{u^A\} = \{P^A\} , \tag{5}$$

can be transformed using Equation 3 so that

$$[M^B]\{\ddot{u}^B\} + [B^B]\{\dot{u}^B\} + [K^B]\{u^B\} = \{P^B\} , \tag{6}$$

where

$$[M^B] = [G^{AB}]^T[M^A][G^{AB}] , \tag{7}$$

$$[B^B] = [G^{AB}]^T[B^A][G^{AB}] , \tag{8}$$

$$[K^B] = [G^{AB}]^T[K^A][G^{AB}] , \tag{9}$$

and

$$\{P^B\} = [G^{AB}]^T\{P^A\} . \tag{10}$$

After the solution $\{u^B\}$ is obtained, data recovery for $\{u^A\}$ is accomplished using the equation

$$\{u^A\} = \left\{\begin{array}{c} u_b^A \\ \hline u_i^A \end{array}\right\} = [G^{AB}]\{u^B\} + \left\{\begin{array}{c} 0 \\ \hline u_i^o \end{array}\right\} , \tag{11}$$

where

$$\{u_i^o\} = [K_{ii}]^{-1}\{P_i^A\} . \tag{12}$$

The vector $\{P_i^A\}$ represents the applied loads on the interior points and is a partition of the total load vector $\{P^A\}$. The vector $\{u_i^o\}$ contains the additional interior point deflections due to applied loads on interior points.

The Guyan reduction procedure as used in eigenvalue analysis implies that the $\{u_i^o\}$ vector is not used (see Section 3.5.4). The formulation for the transformation matrix $[G^{AB}]$ implies that the boundary set, $\{u_b^A\}$, for substructure A, should be selected so that $[K_{ii}]$ is non-singular. In other words, boundary degrees of freedom must be selected so that interior degrees of freedom do not define free-body motion. The matrix $[G^{AB}]$ is denoted as the HØRG matrix for statically reduced substructure A stored on the substructure data base file as described in Section 1.10.2 of the User's Manual.

4.6.2 Modal Matrix Reduction

The modal synthesis capabilities of automated multi-stage substructure analysis are facilitated through use of modal reduce operations. Using modal reduction, a substructure is reduced from one set of coordinates to a new, smaller number of degrees of freedom consisting of user-selected boundary displacement coordinates and modal coordinates. Modal reduction may be performed on real, symmetric mass and stiffness matrices (real modal reduction), and on complex or unsymmetric matrices, or on systems with damping (complex modal reduction). A "user mode" option is provided so that the user may also define a substructure in terms of modal data obtained external to NASTRAN by test or analysis.

The complete theoretical development of the modal transformation matrices is presented in Section 4.7. A summary of the modal reduction transformation matrices is presented in the sections which follow. The "user mode" option is described in Section 4.7.4.

4.6.2.1 Real Modal Reduction Transformations

As before for static reduction, assume substructure A is to be reduced and substructure B is the new, reduced structure. $\{u_b^A\}$ are the retained boundary degrees of freedom and $\{u_i^A\}$ are the interior degrees of freedom to be reduced.

The transformation from $\{u^A\}$ to $\{u^B\}$ is written as

$$\{u^A\} = [G^{AB}]\{u^B\} , \tag{13}$$

where

$$\{u^A\} = \left\{\begin{array}{c} u_b^A \\ \hdashline u_i^A \end{array}\right\}, \tag{14}$$

$$[G^{AB}] = \left[\begin{array}{c:c:c} I & 0 & 0 \\ \hdashline G_{ib} & H_{io} & \phi_i - G_{ib}\phi_b \end{array}\right], \tag{15}$$

and

$$\{u^B\} = \left\{\begin{array}{c} u_b^B \\ \hdashline \delta_o^B \\ \hdashline \delta^B \end{array}\right\} . \tag{16}$$

In the above equations, $\{u_b^B\}$, are the physical coordinates for substructure B, $\{\delta_o^B\}$ are six inertia relief shape, free-body acceleration coordinates, and $\{\delta^B\}$ are the flexible-mode coordinates. The matrix $[\phi_i]$ contains the flexible mode shapes for the interior degrees of freedom (including null rows if interior coordinates were constrained for mode extraction), and $[\phi_b]$ are the flexible mode shapes for those boundary coordinates not constrained for mode extraction. In addition, as in static reduction using the matrix partitions of Equation 1,

$$[G_{ib}] = -[K_{ii}]^{-1}[K_{ib}] . \tag{17}$$

The transformation partition for inertia relief shapes is

$$[H_{io}] = -[K_{ii}]^{-1}([M_{ib}] + [M_{ii}][G_{ib}])[\phi_{bo}] , \tag{18}$$

where $[\phi_{bo}]$ are free-body displacement shapes computed using boundary point coordinate geometry.

The transformation matrix $[G^{AB}]$ is used as shown in Equations 7, 8, 9, and 10 to define substructure B. As before, $[G^{AB}]$ is the HØRG matrix for substructure A.

After solution vectors $\{\ddot{u}^B\}$, $\{\dot{u}^B\}$, and $\{u^B\}$ are obtained, $\{u^A\}$ is computed using

$$\left\{\begin{array}{c} u_b^A \\ \hdashline \bar{u}_i^A \end{array}\right\} = [G^{AB}]\{u^B\} , \tag{19}$$

$$\left\{\begin{array}{c} \dot{u}_b^A \\ \hdashline \dot{u}_i^A \end{array}\right\} = [G^{AB}]\{\dot{u}^B\} , \tag{20}$$

and

$$\left\{\begin{array}{c} \ddot{u}_b^A \\ \hdashline \ddot{u}_i^A \end{array}\right\} = [G^{AB}]\{\ddot{u}^B\} . \tag{21}$$

The standard procedure to define substructure A interior coordinate displacements is defined as

$$\{u_i^A\} = [K_{ii}^A]^{-1}\{P_i^A\} + \{\bar{u}_i^A\} \tag{22}$$

However, a mode acceleration technique for improved accuracy may be requested by the user such that

$$\{\Delta u_i^A\} = [K_{ii}^A]^{-1}\Big(\{P_i^A\} - [M_{ib}^A]\{\ddot{u}_b^A\} - [M_{ii}^A]\{\ddot{u}_i^A\} - [B_{ib}]\{\dot{u}_b^A\} - [B_{ii}]\{\dot{u}_i^A\}\Big) . \qquad (23)$$

Note that the mode acceleration technique may be costly for transient analyses.

The improved total deflections are computed using

$$\{u_i^A\} = [G_{ib}]\{u_b^A\} + \{\Delta u_i^A\} . \qquad (24)$$

4.6.2.2 Complex Modal Reduction Transformations

Structural models defined by unsymmetric matrices, matrices with complex terms, or damping matrices may be reduced using complex modal reduction.

Using the same notation developed in the previous section for real modal reduction, a displacement transformation matrix is written as

$$\{u^A\} = [G^{AB}]\{u^B\} , \qquad (25)$$

where

$$\{u^A\} = \left\{\begin{array}{c} u_b^A \\ \hline u_i^A \end{array}\right\} , \qquad (26)$$

$$[G^{AB}] = \left[\begin{array}{c|c} I & 0 \\ \hline G_{ib} & \psi_i - G_{ib}\psi_b \end{array}\right] , \qquad (27)$$

and

$$u^B = \left\{\begin{array}{c} u_b^B \\ \hline \chi^B \end{array}\right\} . \qquad (28)$$

In general, the mode shapes $[\psi_i]$ and $[\psi_b]$ are complex. $[G_{ib}]$ is the static reduction transformation and $\{\chi^B\}$ are the complex, flexible mode, generalized coordinates for substructure B. Inertia relief coordinates are not used for complex modal reduction.

A second transformation matrix is required for transformation of forces in the unsymmetric case. It is defined as

$$[\bar{\bar{G}}^{AB}] = \left[\begin{array}{c|c} I & 0 \\ \hline \bar{\bar{G}}_{ib} & \bar{\bar{\psi}}_i - G_{ib}\bar{\bar{\psi}}_b \end{array}\right] , \qquad (29)$$

where $$[\hat{G}_{ib}] = -[K^T_{ii}]^{-1}[K_{bi}]^T , \qquad (30)$$

and $[\bar{\bar{\psi}}]$ are the left-hand eigenvectors obtained by solving the eigenvalue problem using the transpose of the mass, damping, and stiffness matrices of substructure A.

The matrices for substructure B are obtained using

$$[M^B] = [\bar{\bar{G}}^{AB}]^T[M^A][G^{AB}] , \qquad (31)$$

and similarly for the damping and stiffness matrices, and the forces are transformed using

$$\{P^B\} = [\bar{\bar{G}}^{AB}]^T\{P^A\} . \qquad (32)$$

Matrix $[G^{AB}]^T$ is used for $[\bar{\bar{G}}^{AB}]^T$ when the matrices of substructure A are symmetric.

Matrix $[G^{AB}]$ is the HØRG matrix for substructure A and $[\bar{\bar{G}}^{AB}]$ is the HLFT matrix item for substructure A stored on the substructure data base file.

4.6.3 Substructure Combinations

The mathematical operations involved with combining substructures are very similar to the methods of combining element stiffness matrices in the finite element displacement method. For each element or substructure, the matrices are generated in a local coordinate system, they are then transformed to a common global coordinate system, expanded to the size of the combination matrix, and added together. If adjacent elements or substructures are connected at the same grid point, the combination is effected by simply adding the corresponding contributions from each matrix.

In substructure analysis, the transformations between the displacements in each substructure $\{u^A\}$, $\{u^B\}$, etc., and those in the combination, $\{u^C\}$, are given by a set of matrices $[H^{AC}]$, $[H^{BC}]$, etc., where:

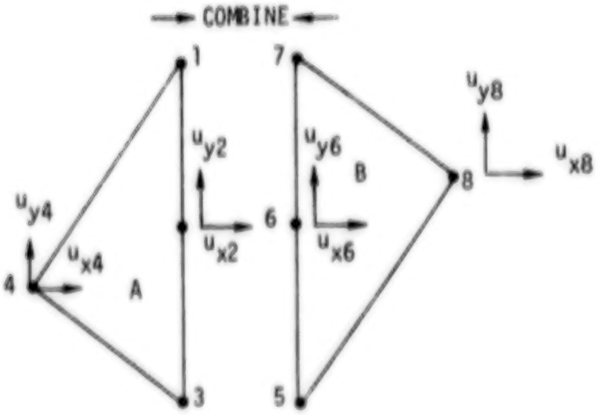
$$\begin{aligned} \{u^A\} &= [H^{AC}]\{u^C\} \\ \{u^B\} &= [H^{BC}]\{u^C\} \end{aligned} \tag{33}$$

etc.

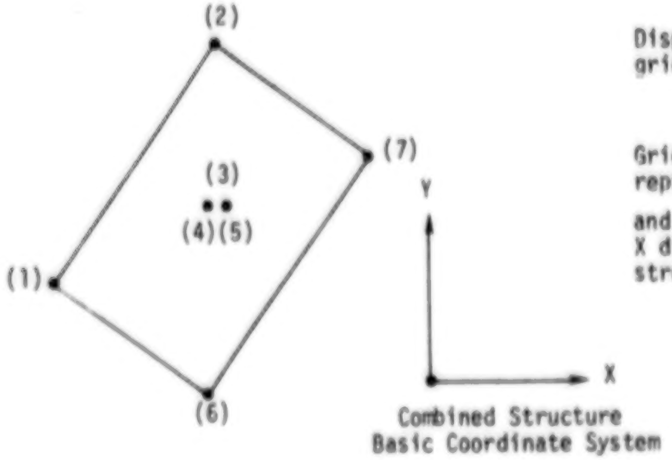
The contents of these matrices may be illustrated by the example in Figure 1. Two substructures, A and B, must be combined using the degrees of freedom along the boundary which are sequenced differently for each substructure. Note that grid points 2 and 6 of the original structures are to be connected only in their x-component of displacement. The final desired sequence is shown in parentheses. For simplicity, only two degrees of freedom are given for each grid point. In general, however, six-by-six matrix partitions would be used in place of these unit values.

If $[K^A]$ and $[K^B]$ are the stiffness matrices for substructures A and B, the stiffness matrix for the combination C is:

$$[K^C] = [H^{AC}]^T[K^A][H^{AC}] + [H^{BC}]^T[K^B][H^{BC}] \ . \tag{34}$$



$$\{u^A\} = \begin{Bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \\ u_{x4} \\ u_{y4} \end{Bmatrix} \text{ and } \{u^B\} = \begin{Bmatrix} u_{x5} \\ u_{y5} \\ u_{x6} \\ u_{y6} \\ u_{x7} \\ u_{y7} \\ u_{x8} \\ u_{y8} \end{Bmatrix}$$



Combined Structure
Basic Coordinate System

Displacement component in Y direction at grids 2 and 6 are not to be connected.

Grid points 4 and 5 independently represent Y displacements (u_{y2} and u_{y6}) and grid point 3 represents the combined X displacement (u_{x2} and u_{x6}) in final structure.

$$\{u^C\} = \begin{Bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y4} \\ u_{y5} \\ u_{x6} \\ u_{y6} \\ u_{x7} \\ u_{y7} \end{Bmatrix}$$

$$\{u^A\} = [H^{AC}]\{u^C\} \text{ where } [H^{AC}] = \begin{array}{c|ccccccccccc} \text{DOF} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\{u^B\} = [H^{BC}]\{u^C\} \text{ where } [H^{BC}] = \begin{array}{c|ccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Figure 1. Illustration of substructure combination matrices.

The mass and damping matrices would be combined with identical transformations. The load vectors on the combination structure are:

$$\{P^C\} = [H^{AC}]^T \{P^A\} + [H^{BC}]^T \{P^B\} \tag{35}$$

The contributions to $\{P^C\}$ from each basic substructure are not added at this time. They are stored separately with the data for each combined substructure. At solution time, the user has the option to specify arbitrary linear combinations of these loadings in order to define the total loading to be applied to the model. The solution vectors $\{u^A\}$ and $\{u^B\}$ for the contributing substructures are recovered using Equation 33 once the displacements or mode shapes $\{u^C\}$ have been computed.

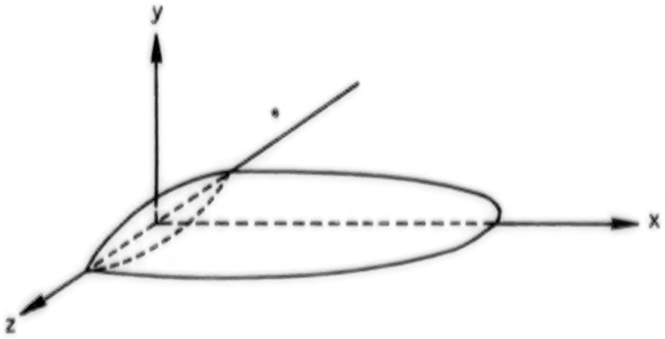
This method of performing the substructure combinations offers the following advantages:

1. The $[H^{AC}]$ matrices are extremely sparse, they require a minimum of storage space and the NASTRAN matrix multiply operations used are efficient.
2. For each substructure, the same transformation is applied to all its matrices, loads and solution vectors.
3. If rotations and/or symmetry transformations are specified for the component substructures, the directions of displacement may be transformed by simply substituting direction cosines instead of unit values, as discussed below.

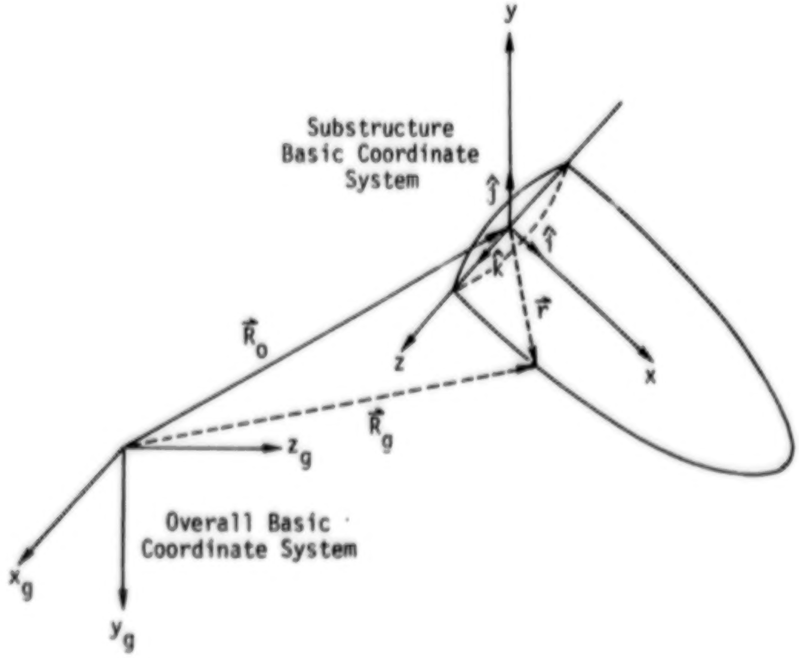
In many situations when structures are to be combined, it is necessary to translate and/or rotate a component substructure in order to bring its boundary points into alignment with the other substructures. The user may define the physical orientation of each component substructure by specifying the orientation of the substructure basic coordinate system relative to the basic coordinate system of the combined structure, herein defined as the overall basic coordinate system. A symmetric mirror image of the substructure may also be obtained by specifying the axis normal to the plane of reflection desired. This allows the user to prepare only one half of a symmetric model and automatically obtain the matrices for the other half via this symmetric transformation.

Consider the horizontal stabilizer illustrated below as a component substructure defined in its own basic x, y, z coordinate system:

ORIGINAL PAGE IS OF POOR QUALITY



This stabilizer must now be rotated, translated, and attached to another substructure representing the fuselage. The user defines the location and orientation of the stabilizer coordinate system in terms of the overall basic coordinate system. This is illustrated in the following sketch of the overall basic coordinate system x_g, y_g, z_g.


y
Substructure
Basic Coordinate
System
$\hat{j}$
$\hat{i}$
$\hat{k}$
$\vec{r}$
$\vec{R}_o$
z
x
z_g
$\vec{R}_g$
Overall Basic
Coordinate System
x_g
y_g

The locations of the substructure points in the overall basic coordinate system are described by the equation:

$$\{R_g\} = [T_{tr}] \{r\} + \{R_o\} \tag{36}$$

where $[T_{tr}]$ is a rotation matrix of unit vectors defined by the user such that:

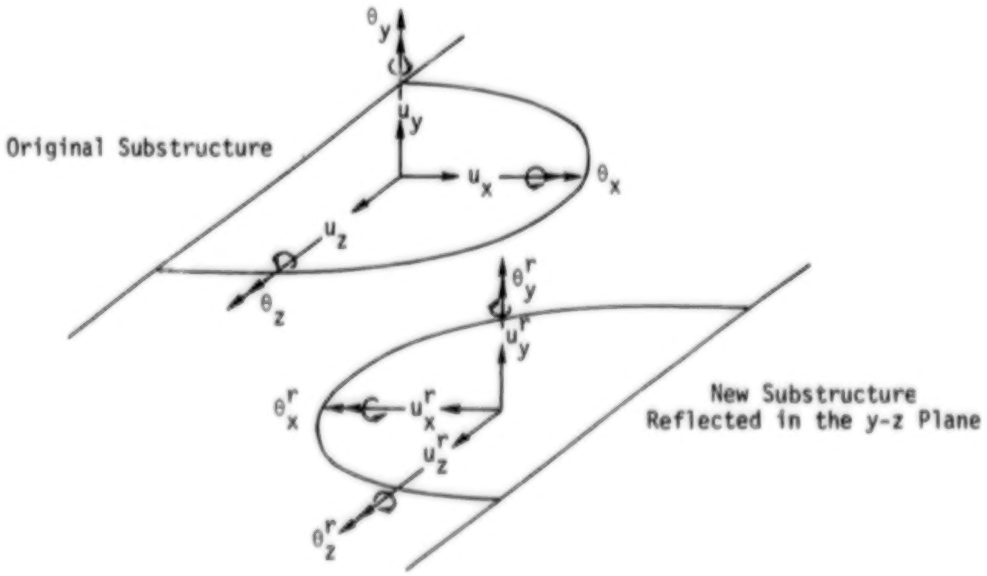
$$[T_{tr}] = \begin{bmatrix} i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ i_3 & j_3 & k_3 \end{bmatrix} \qquad (37)$$

and R_0 is the offset vector from the overall basic origin to the origin of the stabilizer coordinate system.

The displacement vector $\{u_g\}$ for a grid point in the overall basic coordinate system is similarly defined in terms of a vector $\{u_s\}$ for the component substructure in its basic coordinate system by the equation:

$$\{u_g\} = [T_{tr}]\,\{u_s\} \qquad (38)$$

The symmetric reflection of a substructure is treated similarly. Note that the mirror image model has identical stiffness and mass matrices, however, the components of displacement and rotation are now defined in a left-handed coordinate system. Therefore, a corresponding transformation is required to provide for compatability among combined substructures. The following sketches illustrate the problem of a structure reflected in the overall basic z-y plane:


θ_y
u_y
Original Substructure
u_x
θ_x
u_z
θ_z
θ^r_y
u^r_y
New Substructure
Reflected in the y-z Plane
θ^r_x
u^r_x
u^r_z
θ^r_z

Both the displacement and rotation components, prior to transformation, are now in a left-handed system. Therefore, in order to connect the reflected structure to the original model with its right-handed coordinate system, the following transformation must be applied to every grid point:

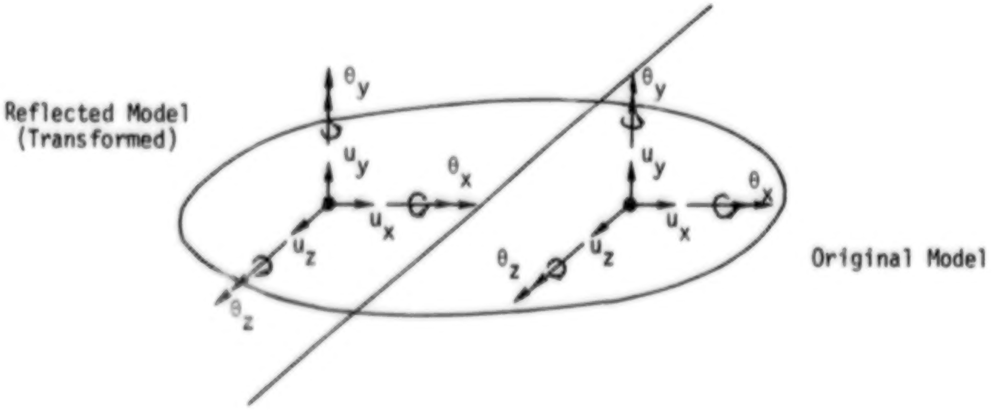
$$\{u_g\} = \begin{Bmatrix} u^r_x \\ u^r_y \\ u^r_z \\ \theta^r_x \\ \theta^r_y \\ \theta^r_z \end{Bmatrix} = [T_{sym}] \begin{Bmatrix} u_x \\ u_y \\ u_z \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} \qquad (39)$$

where, for the example shown:

$$[T_{sym}] = \begin{bmatrix} -1 & & & & & 0 \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & -1 & \\ 0 & & & & & -1 \end{bmatrix} \qquad (40)$$

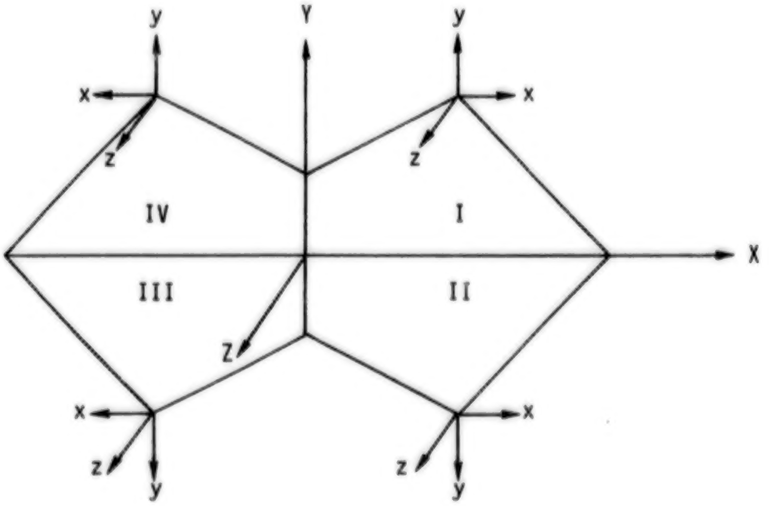
Similar transformations exist for reflections in the x-z and x-y planes.

Assuming the primary substructure and its reflected model are to be combined along the plane of symmetry, the resulting combined model would show:


Reflected Model
(Transformed)
θ_y
u_y
θ_x
u_x
u_z
θ_z
θ_y
u_y
θ_x
u_x
θ_z
u_z
Original Model

All subsequent combinations using this new substructure would proceed as for any other substructure. On data recovery, however, the user must be careful. The detailed results obtained for the basic reflected model (Phase 3) will be output in the reflected coordinate system and, therefore, must be interpreted with the left-hand rule. These results, of course, include not only displacements and rotations at grid points, but they also include stresses and element forces and moments.

If a subsequent symmetry transformation is imposed on a model already comprising components which themselves resulted from symmetry transformations, a left-handed coordinate system of the earliest component substructure would again be transformed, but now to a right-handed system. For example, a model with four identical components could be constructed as follows:



where the original basic substructure (I) was reflected in the x-z plane to create substructure (II). Substructures (I) and (II) were combined and subsequently reflected in the y-z plane to create substructures (III) and (IV) to complete the four part model. The final model would comprise four basic substructures, of which substructures II and IV would produce Phase 3 results in the left-handed coordinate systems as shown.

The transformations and symmetric reflections of the structural matrices are easily included in the process of combining several substructures. The combined transformation equations necessary to define displacements and rotations of each grid point $\{u_s\}$, in basic, right-handed, coordinates are:

$$\{u_g\} = [H_g]\{u_s\} \qquad (41)$$

where

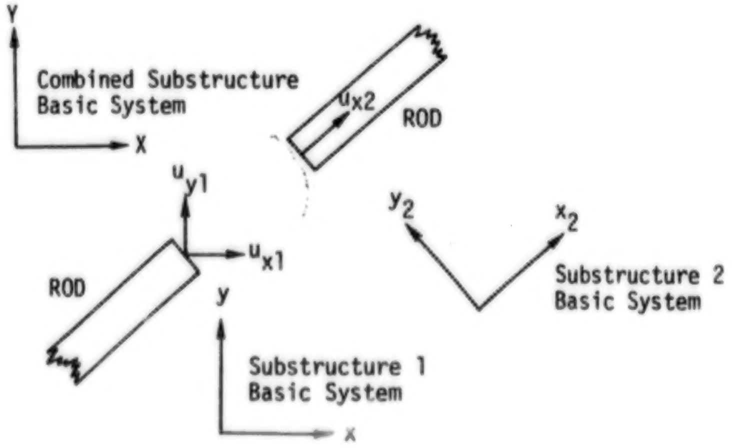
$$[H_g] = [T_{sym}] \left[\begin{array}{c:c} T_{tr} & \\ \hdashline & T_{tr} \end{array}\right] \qquad (42)$$

and $\{u_s\}$ are the displacements in the respective component substructure coordinate systems.

The $[H_g]$ matrix for each grid point is assembled into the overall substructure transformation matrix $[H^{AC}]$ defined previously. The stiffness matrix, mass matrix, and load vectors can now be transformed to correspond to displacements and rotations in the overall basic coordinate system of the combination structure. The $[H^{AC}]$ matrix is the H0RG matrix item for substructure A stored on the substructure data base file.

Several restrictions and rules must be imposed on these transformations in order to prevent complications and errors. The transformations given above will be applied only to grid points having their displacements defined in the original substructure basic coordinate system. Displacements defined in a local system of the component substructure will not be transformed, i.e., their directions will be fixed on the substructure and will travel with the substructure during its rotation and/or translation. Provisions are made to allow the user to define new local systems, or to change local systems to basic during Phase 2 processing.

If constraints have been applied during Phase I or if selected degrees of freedom at a grid point have been reduced out of the matrices, additional problems may occur. These grid points might not have all of the original three displacements and three rotational degrees of freedom. The transformation matrix (H_g) at these grid points could then reintroduce previously eliminated degrees of freedom which could cause singularities. Consider the example shown in the following sketch where two co-linear rods are connected.


Y
Combined Substructure Basic System
X
u_{x2}
ROD
u_{y1}
y_2
x_2
u_{x1}
ROD
y
Substructure 2 Basic System
Substructure 1 Basic System
x

Assume the component u_{y2} on the right structure was constrained out to remove a singularity. After rotation there would again be two components, u_x and u_y, and the grid point will again be singular. The user has two options to avoid this problem. Multipoint constraints may be imposed during the solve step in Phase 2 if the components are left unconnected until the solution model has been completed. Alternately, the user may define compatible local coordinate systems at both grid points and the desired connections can be made since local grid point coordinate systems are not transformed when the structure is rotated.

4.6.4 Solution and Recovery

After all of the necessary substructures have been assembled and reduced with the operations described above, the assembled pseudostructure may be analyzed with many of the standard NASTRAN options. Static, dynamic, and normal modes analysis are processed automatically requiring only a few special data cards to define the constraints and load combinations with reference to the original basic substructure names. The mathematical steps involved with the solution are described in Section 3.

The results of the NASTRAN solution are in the form of vectors related to the pseudostructure degrees of freedom. They are:

1. Displacement vectors, $\{u_g\}$, with one vector per solution load case in static analysis, time point in transient analysis, and frequency in frequency response analysis. In normal modes analysis, one displacement vector is produced for each eigenvector obtained during solution.

2. Forces of constraint, $\{q\}$, with one vector per solution load case, time, frequency, or eigenvector. These vectors are computed differently than NASTRAN single-point forces of constraint as described later in this section.

3. For static or dynamic analysis, a set of load vectors, $\{P_g\}$, are produced.

4. In normal modes analysis, an eigenvalue, λ_i, the natural frequency, f_i, a modal mass, M_i, and a modal stiffness, K_i, are produced for each mode shape.

The recovery of displacement vectors for the original basic substructures are obtained by tracing backwards the same path that created the structural matrices and loads. For instance, if substructure A were reduced to produce substructure B with the transformation matrix $[G^{AB}]$ as defined in Section 4.6.1, the displacements of structure A are obtained by

$$\{u^A\} = [G^{AB}]\{u^B\} + \{u_i^o\} . \tag{43}$$

Note that $\{u_i^o\}$, as defined by Equation 12, is not produced for normal modes analysis.

Continuing the example, assume that substructure F was created by combining substructures A and B. The displacements of these substructures may be obtained by application of Equation 33:

$$\{u^A\} = [H^{AF}]\{u^F\} \tag{44}$$

$$\{u^B\} = [H^{BF}]\{u^F\} \tag{45}$$

where $[H^{AF}]$ and $[H^{BF}]$ were created originally in order to combine these two substructures.

The forces of constraint, $\{q\}$, play a different role in substructure analysis than in non-substructuring NASTRAN formulations. The total set of forces on an individual substructure are computed in Phase 2 execution as shown below and are identified by rigid format application. In these equations, $\{q\}$ are the forces of constraint, $\{P\}$ are the applied loads, $\{u\}$ is the displacement vector, $[K]$ is the stiffness, $[B]$ is the damping, $[M]$ is the mass, ω^2 are eigenvalues from a real modes analysis, and p are complex eigenvalues from a complex modal reduction.

Rigid Format	Equation
1 and 2	$\{q\} = [K]\{u\} - \{P\}$
3 (real)	$\{q\} = [K]\{u\} - [M]\left\lceil \omega^2 \right\rfloor\{u\}$
3 (complex)	$\{q\} = [K]\{u\} + [B]\left\lceil p \right\rfloor\{u\} + [M]\left\lceil p^2 \right\rfloor\{u\}$
8 and 9	$\{q\} = [K]\{u\} + [B]\{\dot{u}\} + [M]\{\ddot{u}\} - \{P\}$

The force vectors $\{q\}$ contain all the terms due to:

1. Applied forces
2. Inertia and damping forces
3. Single-point constraints
4. Multipoint constraints
5. Forces transferred from other connected substructures
6. Residual forces due to computer round-off

MISCELLANEOUS GENERAL PROVISIONS FOR STRUCTURAL ANALYSIS

The additional solution quantities such as element forces and stresses are calculated in Phase 3 using the equations in NASTRAN for each rigid format application.

4.7 COMPONENT MODE SYNTHESIS

Component mode synthesis shares with substructure analysis the concept of dividing a structure into separate components, reducing the order of the component matrices, and combining the resultant matrices through the displacement equalities at the common boundary points. Modal synthesis has the primary advantage that the dynamic response of a substructure may be represented by a small number of modal displacements which may be more accurate than an arbitrary set of reduced grid displacements. A secondary advantage is that modal synthesis may be used as a design tool in that the participation of critical component modes in the overall system response are easily obtained.

In the following development, both real normal modes and complex eigenvectors will be used to formulate the reduced system. The undamped mode shapes, or real normal modes, are used to represent the unconnected parts of a component structure. In the complex eigenvalue problem, the general case of unsymmetric matrices with damping and complex matrix terms is considered. The primary theoretical development concentrates on the real modes problem with the complex eigenvector formulation treated as a more general extension.

The NASTRAN system provides a large variety of structural and nonstructural applications. Similarly, the modal synthesis method is a general capability. Because of potentially large-size structural models, it is also designed to be efficient and versatile.

The modal synthesis capability has the following basic characteristics:

1. The system is compatible with the automated multi-stage substructure (AMSS) system.
2. No internal restrictions on the type of modes (free-free, fixed, etc.) are imposed.
3. Connection between modal synthesis substructures and normal substructures is allowed.
4. User-specified modal data is allowed.
5. Complex eigenvalue modal synthesis capability is available.

The equations for the modal synthesis formulation and data recovery are developed in this section. The associated substructure operations such as combining substructure components and providing overall solutions are described in Section 4.6.2.

The following development is divided into four parts. The transformations for the case of real normal modes is developed in detail. Then, a brief development extends this method to the case of complex eigenvalues for damped systems. Next, the data recovery equations are provided, including the energy factor calculations for the component modes, which help the user measure solution accuracy. The final section describes the methodology for processing user-supplied modal data. This modal data may be derived from structure testing or from analyses by other methods.

4.7.1 Real Modal Reduction

This section presents the development of the equations for the real, normal mode, modal reduction of a substructure. This modal reduction is applicable to undamped or lightly damped structures represented by symmetric mass and stiffness matrices. The Substructure Control Deck command MREDUCE controls execution of the real mode modal reduction.

4.7.1.1 Real Mode Transformation

Each component substructure is defined by a stiffness matrix, $[K]$, a mass matrix, $[M]$, and optional damping matrices, $[B]$ and $[K^4]$. For the undamped case, the equations of motion are

$$[M]\{\ddot{u}\} + [K]\{u\} = \{P\} , \tag{1}$$

where $\{u\}$ are the displacement coordinates and $\{P\}$ are the loads due to external forces or boundary reaction loads from other substructures.

For the idealized case when all mode shape vectors, $\{\phi_j\}$, of the unconstrained substructure have been extracted, the equations of motion are uncoupled, one equation for each mode, such that

$$\mu_j(\ddot{\xi}_j + \omega_j^2\xi_j) = P_j , \tag{2}$$

where

$$\mu_j = \{\phi_j\}^T[M]\{\phi_j\} , \tag{3}$$

$$\omega_j^2 = (1/\mu_j)\{\phi_j\}^T[K]\{\phi_j\} , \tag{4}$$

and

$$P_j = \{\phi_j\}^T\{P\} . \tag{5}$$

Also,

$$u_i = \sum_{j=1}^{N} \phi_{ij}\xi_j \quad \text{or} \quad \{u\} = [\phi]\{\xi\} . \tag{6}$$

In this development, j denotes the mode, i denotes a displacement coordinate, and N is the total number of modes. The matrix $[\phi]$ represents the collection of all mode shape vectors, and $[\phi_j]$ represents the set of mode shape vectors to be retained in the analysis.

Using the Laplace operator s where $\partial/\partial t = s$, Equations 2, 5, and 6 are combined to obtain the matrix equation for coordinate displacements so that

$$\{u\} = [\phi]\left[\ulcorner\frac{1}{\mu_j(s^2+\omega_j^2)}\lrcorner\right][\phi]^T\{P\} . \tag{7}$$

Equation 7 is exact when all modes of the system are used. Normally, however, only a truncated set, $\xi_1 \ldots \xi_m$, is used to reduce the number of variables.

4.7.1.2 Effects of Truncated Modes

The response of the higher modes, $\{\phi_k\}$, where $\omega_k^2 >> s^2$ is primarily static, and Equation 7 becomes

$$\{u\} = [\phi_j]\left[\ulcorner Z_j(s)\lrcorner\right][\phi_j]^T\{P\} + [\phi_k]\left[\ulcorner Z_k\lrcorner\right][\phi_k]^T\{P\} , \tag{8}$$

where

$$Z_j(s) = \frac{1}{\mu_j(s^2+\omega_j^2)} \quad \text{for } j \leq m , \tag{9}$$

and

$$Z_k = \frac{1}{\mu_k\omega_k^2} = \frac{1}{K_k} \quad \text{for } k > m , \tag{10}$$

and the number of retained modes is m.

The form of the last term of Equation 8 is that of a flexibility matrix, $[\Delta Z]$, times a load, which produces a corrective displacement vector. This residual flexibility is obtained from the basic equations of motion, Equation 1, as shown next.

Equation 8 may be written in terms of free-body modes $[\phi_o]$, retained flexibility modes $[\phi_j]$, and residual flexibility in the form

$$\{u\} = \left[\frac{1}{s^2}[\phi_o]\left[\ulcorner\mu_o\lrcorner\right]^{-1}[\phi_o]^T + [\phi_j]\left[\ulcorner Z_j(s)\lrcorner\right][\phi_j]^T + [\Delta Z]\right]\{P\} , \tag{11}$$

where the subscript, o, denotes a free-body mode.

For constant applied loads, only free-body acceleration exists, and

$$\{\ddot{u}\} = s^2\{u\} = [\phi_o]\left[\ulcorner\mu_o\lrcorner\right]^{-1}[\phi_o]^T\{P\} \qquad (s \rightarrow 0) . \tag{12}$$

Substituting Equations 11 and 12 into Equation 1 with $s = 0$, and noting that $[K]\{\phi_o\} = 0$, we obtain

$$[K][\Delta Z]\{P\} = \{P\} - [M][\phi_o][\mu_o]^{-1}[\phi_o]^T\{P\} - [K][\phi_j][Z(0)][\phi_j^T]\{P\} . \tag{13}$$

Multiplying Equation 11 by [K] and substituting Equation 13 into that result gives an approximation, below, which includes the static effect of all modes and the dynamic effect of selected modes

$$[K]\{u\} \approx [K][\phi_j][Z(s) - Z(0)][\phi_j^T]\{P\} - [M][\phi_o][\mu_o]^{-1}[\phi_o]^T\{P\} + \{P\} . \tag{14}$$

The above approximation would provide excellent results for an uncoupled structure. However, at the boundary points, the loads P gain contributions from the connected structures due to the displacements at the boundary. For this reason, the boundary degrees of freedom must be separated from the interior degrees of freedom in order to proceed. But first, Equation 14 will be simplified by introducing modal coordinates, $\{\delta\}$, as defined below. Let

$$[K]\{u\} = [K][\phi_j]\{\delta\} - [M][\phi_o]\{\delta_o\} + \{P\} , \tag{15}$$

where

$$\{\delta\} = [Z(s) - Z(0)][\phi_j]^T\{P\} , \tag{16}$$

and

$$\{\delta_o\} = [\mu_o]^{-1}[\phi_o]^T\{P\} . \tag{17}$$

Note that $\{\delta\}$ represents the difference between the modal displacement and the static response. Substituting Equation 9 into Equation 16, each modal coordinate is obtained as

$$\delta_j = -\frac{s^2[\phi_j]^T\{P\}}{\mu_j\omega_j^4(1 + s^2/\omega_j^2)} . \tag{18}$$

The quantity δ_j has units of displacement and decreases as the fourth power of ω_j. The response magnitude of the high frequency modes therefore converges rapidly. Also note that δ_o has units of acceleration and represents free-body motion.

4.7.1.3 Interior/Boundary Transformations

For connection to other structures, Equation 15 must be separated into boundary points, u_b, and interior points, u_i. For example, the matrices are partitioned such that:

$$[K] \rightarrow \left[\begin{array}{c|c} K_{bb} & K_{bi} \\ \hline K_{ib} & K_{ii} \end{array}\right] , \tag{19}$$

and

$$u \rightarrow \left\{\begin{array}{c} u_b \\ \hline u_i \end{array}\right\} \text{ , etc.} \tag{20}$$

Although the full matrix [K] may be singular, the partition $[K_{ii}]$ must be non-singular. The interior displacements may be defined by the lower half of Equation 15 which becomes

$$\begin{aligned} [K_{ib}]\{u_b\} + [K_{ii}]\{u_i\} &= [K_{ib}\phi_b + K_{ii}\phi_i]\{\delta\} \\ &- [M_{ib}\phi_{bo} + M_{ii}\phi_{io}]\{\delta_o\} + \{P_i\} . \end{aligned} \tag{21}$$

Solution of Equation 21 for the interior displacements, $\{u_i\}$, yields the result

$$\begin{aligned} \{u_i\} &= [G_{ib}]\{u_b\} + [\phi_i - G_{ib}\phi_b]\{\delta\} \\ &- [K_{ii}]^{-1}[M_{ib}\phi_{bo} + M_{ii}\phi_{io}]\{\delta_o\} + [K_{ii}]^{-1}\{P_i\} , \end{aligned} \tag{22}$$

where

$$[G_{ib}] = - [K_{ii}]^{-1}[K_{ib}] . \tag{23}$$

Also, note that $\{\phi_{io}\} = [G_{ib}]\{\phi_{bo}\}$ for the rigid body modes and the matrix $[G_{ib}]$ is identical to the transformation matrix used for the Guyan reduction technique. Without modal coordinates, the formulation degenerates to a static matrix condensation.

A higher order approximation, $\{\tilde{u}_i\}$, to the interior point displacements is obtained as follows. An expression for $\{\ddot{u}_i\}$ is obtained by differentiating Equation 22. Equation 1 is rewritten in partitioned form and $\{\tilde{u}_i\}$ is used instead of $\{u_i\}$ so that

$$\left[\begin{array}{c|c} M_{bb} & M_{bi} \\ \hline M_{ib} & M_{ii} \end{array}\right] \begin{Bmatrix} \ddot{u}_b \\ \ddot{u}_i \end{Bmatrix} + \left[\begin{array}{c|c} K_{bb} & K_{bi} \\ \hline K_{ib} & K_{ii} \end{array}\right] \begin{Bmatrix} u_b \\ \tilde{u}_i \end{Bmatrix} = \begin{Bmatrix} P_b \\ P_i \end{Bmatrix} . \tag{24}$$

The second derivative of Equation 22 is substituted into the lower expression of Equation 24, and the expression is solved for $\{\bar{u}_i\}$, yielding

$$
\begin{aligned}
\{\bar{u}_i\} = {} & [G_{ib}]\{u_b\} - [K_{ii}]^{-1}[M_{ib} + M_{ii}G_{ib}]\{\ddot{u}_b\} \\
& - [K_{ii}]^{-1}[M_{ii}]\Big[[\phi_i - G_{ib}\phi_b]\{\ddot{\delta}\}\Big] \\
& + [K_{ii}]^{-1}[M_{ii}][K_{ii}]^{-1}[M_{ib} + M_{ii}G_{ib}][\phi_{bo}]\{\ddot{\delta}_o\} \\
& + [K_{ii}]^{-1}[P_i - M_{ii}K_{ii}{}^{-1}\ddot{P}_i] \, . \qquad (25)
\end{aligned}
$$

Equation 25 is substituted into the upper expression of Equation 24 to obtain the dynamic equations of motion for the boundary coordinates. In a similar fashion, the equations of motion for the modal coordinates may also be developed resulting in a complete definition of the reduced system. However, the equations of motion resulting from the operations described above are identical to those obtained by application of the Galerkin principle as shown below.

The coordinate transformation matrix, $[H_{gh}]$, is defined using Equation 22 such that

$$
\{u\} = \left\{ \begin{array}{c} u_b \\ \hdashline \bar{u}_i \end{array} \right\} = [H_{gh}]\{u_h\} = [H_{gh}] \left\{ \begin{array}{c} u_b \\ \hdashline \delta_o \\ \hdashline \delta \end{array} \right\} \qquad (26)
$$

where

$$
[H_{gh}] = \left[\begin{array}{c:c:c} I & 0 & 0 \\ \hdashline G_{ib} & H_{io} & \phi_i - G_{ib}\phi_b \end{array} \right] , \qquad (27)
$$

$$
[G_{ib}] = - [K_{ii}]^{-1}[K_{ib}] \, , \qquad (28)
$$

and

$$
[H_{io}] = - [K_{ii}]^{-1}([M_{ib}] + [M_{ii}][G_{ib}])[\phi_{bo}] \, . \qquad (29)
$$

When the number of modes is zero and the inertia relief effects are ignored, the transformation matrix given in Equation 27 is the same as Guyan reduction or matrix condensation transformation. When modes exist, they provide dynamic motion relative to static deformations. Free-body motion and redundant constraint information are contained in the $[G_{ib}]$ transformation. Inertia relief deformation shapes are contained in the $[H_{io}]$ matrix. The generalized coordinates, $\{u_h\}$, define the reduced coordinate set for modal-reduced NASTRAN structures.

After applying the transformations defined by Equation 22, the reduced mass and damping matrices, $[M_{hh}]$ and $[B_{hh}]$, are typically coupled between the boundary and modal coordinates, where

$$[M_{hh}] = [H_{gh}]^T[M][H_{gh}] , \tag{30}$$

and

$$[B_{hh}] = [H_{gh}]^T[B][H_{gh}] . \tag{31}$$

However, the stiffness matrix takes the form

$$[K_{hh}] = \left[\begin{array}{c|c} \bar{K}_{bb} & 0 \\ \hline 0 & K_{pp} \end{array}\right] , \tag{32}$$

where $[\bar{K}_{bb}]$ is the stiffness partition obtained from a Guyan reduction and the subscript p refers to the modal coordinates. Note that the boundary and modal coordinates are statically uncoupled.

The externally applied loads are also transformed, resulting in system loads defined as

$$\{P_h\} = [H_{gh}]^T\{P\} . \tag{33}$$

Examination of Equation 27 and the preceding development indicate the following major features of the method:

1. The interior degrees of freedom of the substructure may be replaced by a much smaller set of modal degrees of freedom. Equation 18 indicates that the effects of the high frequency truncated modes decrease with the fourth power of their natural frequencies.

2. Since the modal coordinates are uncoupled from the boundary grid points, as shown in Equation 32, no approximation is used for static solutions.

3. The type of fixed boundary conditions on the normal modes is completely arbitrary. Either interior modal deflections or boundary deflections $\{\phi_b\}$ may contain zero components.

4. The method gives results equivalent to those of other methods when the corresponding restrictions are applied. Elimination of the inertia relief effects simulates the results of Hurty (Reference 1). In addition, fixing all boundary points results in

exactly the same matrices as those employed by Craig/Bampton (Reference 2). The use of free-free modes with inertia relief effects duplicates Rubin's results (Reference 3).

In summary, this method provides all of the conveniences of the automated multi-stage substructuring (AMSS) system with the additional power and accuracy derived from advanced state-of-the art modal synthesis techniques. Additional features are described in the following sections, which develop the methods used in the complex eigenvalue option, the recovery of data from the final solutions, and the processing of user-supplied normal mode data.

4.7.2 Complex Modal Reduction

An extension of the real mode modal reduction is provided here for the general case of structures described by unsymmetric or complex matrix terms of damping matrices that prohibit a normal mode approximation of structural motion.

The basic approach used in real component-mode synthesis will be applied to the complex eigenvector synthesis. The eigenvectors and eigenvalues are obtained by NASTRAN from the general damped system resulting in a set of complex eigenvalues, p, and eigenvectors, $\{\psi\}$, defined by the equation

$$\Big[[M]p^2 + [B]p + [K] \Big]\{\psi\} = \{0\} . \tag{34}$$

As defined in Section 10.4, the orthogonality condition between the eigenvectors is

$$\{\bar{\psi}_i\}^T\Big[(p_i + p_j)[M] + [B] \Big]\{\psi_j\} = 0, \text{ with } i \neq j . \tag{35}$$

where $\{\bar{\psi}_i\}$ is the so-called left eigenvector, obtained by transposing the matrices in Equation 34. The left eigenvector may be considered the characteristic vector of forces for the root, whereby the right eigenvector is the characteristic vector of displacements.

As is done with real mode reduction, boundary (u_b), interior (u_i), and modal coordinates (χ) are defined, and a displacement coordinate transformation is defined as

$$\{u\} = [H_{gz}]\{u_z\} , \tag{36}$$

where

$$\{u\} = \left\{\begin{array}{c} u_b \\ \hdashline u_i \end{array}\right\}, \tag{37}$$

$$[H_{gz}] = \left[\begin{array}{c:c} I & 0 \\ \hdashline G_{ib} & \psi_i - G_{ib}\psi_b \end{array}\right] , \tag{38}$$

and

$$\{u_z\} = \left\{\begin{array}{c} u_b \\ \hdashline \chi \end{array}\right\} \tag{39}$$

Note that, for complex mode reduction, the inertia relief coordinates, $\{\delta_o\}$, are not used.

In addition, a second transformation, required for forces in the unsymmetric case, is defined as

$$[\bar{H}_{gz}] = \left[\begin{array}{c|c} I & 0 \\ \hline \bar{G}_{ib} & \bar{\psi}_i - \bar{G}_{ib}\bar{\psi}_b \end{array}\right] , \tag{40}$$

where

$$[\bar{G}_{ib}] = -[K_{ii}^T]^{-1}[K_{bi}]^T . \tag{41}$$

The resulting system matrices are transformed to reflect the reduced set of displacements with the equations

$$[M_{zz}] = [\bar{H}_{gz}]^T[M][H_{gz}], \text{ etc. } , \tag{42}$$

and the forces, F, are transformed by the equation

$$\{F_z\} = [\bar{H}_{gz}]^T\{F\} . \tag{43}$$

The resulting solution equation is

$$[M_{zz}]\{\ddot{u}_z\} + [B_{zz}]\{\dot{u}_z\} + [K_{zz}]\{u_z\} = \{F_z\} . \tag{44}$$

With this formulation, the reduced structure may be connected to other substructures at the boundary points with the CØMBINE operation.

4.7.3 Solution Data Recovery

After the matrices and loads are determined for the reduced substructure, they may be used in further substructuring operations. The final combination structure may be analyzed for static, eigenvalue, transient, or frequence response. The resulting solution vectors are obtained in terms of the boundary and modal coordinates. This section describes the procedures necessary to recover the detailed data on the interior displacements and the modal energies by component substructure.

4.7.3.1 Solution Recovery for Reduced Structures

After solution vectors, $\{\ddot{u}_h\}$, $\{\dot{u}_h\}$, and $\{u_h\}$, are obtained for a modal reduced substructure, displacements in the original coordinates are recovered in three steps using the equations previously developed, namely

$$\begin{Bmatrix} u_b \\ \hdashline \bar{u}_i \end{Bmatrix} = [H_{gh}]\{u_h\} \, , \tag{45}$$

and similarly for the velocities and accelerations,

$$\begin{Bmatrix} \dot{u}_b \\ \hdashline \dot{\bar{u}}_i \end{Bmatrix} = [H_{gh}]\{\dot{u}_h\} \, , \tag{46}$$

and

$$\begin{Bmatrix} \ddot{u}_b \\ \hdashline \ddot{\bar{u}}_i \end{Bmatrix} = [H_{gh}]\{\ddot{u}_h\} \, . \tag{47}$$

If exterior loads are applied, a correction to interior coordinate displacements is added, resulting in the total displacements, u_i, defined in Equation 22 which are:

$$\{u_i\} = [K_{ii}]^{-1}\{P_i\} + \{\bar{u}_i\} \, . \tag{48}$$

However, Rubin [Reference 3] has shown that a mode acceleration technique will result in improved accuracy. If this technique is selected, then Equation 25 may be used in the following form for the alternative, "improved," displacements

$$\{\Delta u_i\} = [K_{ii}]^{-1}\{P_i + \Delta P_i\} \tag{49}$$

where

$$\begin{Bmatrix} \Delta P_b \\ \hline \Delta \bar{P}_i \end{Bmatrix} = -[M]\{\ddot{u}\} - [B]\{\dot{u}\} . \tag{49a}$$

The total deflections are

$$\{u_i\} = [G_{ib}]\{u_b\} + \{\Delta u_i\} . \tag{50}$$

4.7.3.2 Modal Energy

Because interior degrees of freedom of the structure are replaced by a smaller set of modal coordinates, use of modal synthesis techniques can be expected to result in some degradation of modal accuracy. The magnitude of the error will be dependent on the characteristics of the component substructure loads which are dependent on both the applied loads and the dynamic forces due to other connected substructures. Therefore, the errors are not easily predicted, but they may be measured only for a particular solution of the entire structure.

A measure of solution accuracy may be obtained by computing modal energy factors for the modal coordinates' responses to a solution. The magnitude of the energy factors for modes retained as part of the substructure description (the included modes) determines their participation and indicates whether they may be removed for subsequent analyses. Also, the energy factors for those calculated modes, which were excluded due to limitations of frequency range or number of modes used, may be estimated to determine if they should have been included to improve the accuracy.

The energy factors described here are obtained via the ENERGY subcommand under the RECØVER command. Energy factors from a normal modes or frequency response solution may be computed for any modal reduced substructure.

The total energy factors are determined by the Rayleigh coefficients which are the ratio of energy contained in the mode versus total energy in the system. However, the actual mode shapes used in the formulation are modified such that motions due to boundary point displacements are removed. Separation of the displacement vector into that component due to the modes, $\{\phi\}$, and the remainder, $\{u_r\}$, results in

$$\{u\} = \{u_r\} + \{\psi\}\delta , \tag{51}$$

where
$$\{\psi\} = \left\{\begin{array}{c} 0 \\ \hline \phi_i - G_{ib}\phi_b \end{array}\right\} . \tag{52}$$

The total potential energy is

$$V = \frac{1}{2}\{u\}^T[K]\{u\} . \tag{53}$$

Therefore, the potential energy contribution of each modal coordinate, δ_j, for each solution vector, $\{u\}$, is

$$\delta V_j = \frac{1}{2}\{u\}^T[K]\{\psi\}\delta_j . \tag{54}$$

Similarly, the kinetic energy contribution is

$$\delta T_j = \frac{1}{2}\{\dot{u}\}^T[M]\{\dot{\psi}\}\delta_j , \tag{55}$$

The definition of the velocity vector depends on the type of solution. For instance, in normal mode analysis, $\dot{\delta}_j = \omega\delta_j$.

The energy factors of the excluded modes are estimated from the excitation provided by the resultant solution vector. The equations used are presented below.

The response of modal coordinate δ_k as expressed in terms of static (s) and dynamic (d) effects is

$$\delta_k = \delta_k^d + \delta_k^s . \tag{56}$$

From Equation 18,

$$\delta_k^d = \frac{- s^2\{\phi_k\}^T\{\bar{P}_j\}}{\mu_k\omega_k^4(1 + s^2/\omega_k^4)} , \tag{57}$$

and
$$\delta_k^s = \frac{\{\phi_k\}^T\{\bar{P}_j\}}{\mu_k\omega_k^2} , \tag{58}$$

where $\{\bar{P}_j\}$ is the total load on the substructure at the j^{th} frequency due to boundary and applied loads. This total load is calculated from the equation

ORIGINAL PAGE IS
OF POOR QUALITY

$$\{\bar{P}_j\} = [Ms_j^2 + Bs_j + K]\{u_j\} . \tag{59}$$

For transient analysis, evaluation of Equation 57 would require costly, explicit integration of each coordinate versus time. Therefore, this output is restricted to eigenvalue and frequency response problems where $s = i\omega$ (or $s = p$ for complex eigenvalues).

The expressions for kinetic and potential energy for the excluded modes would normally be written as

$$\delta T_k = \frac{1}{2}\mu_k s^2(\delta_k)^2 , \tag{60}$$

and

$$\delta V_k = \frac{1}{2}\mu_k\omega_k^2(\delta_k)^2 . \tag{61}$$

Substituting Equation 56 into Equations 60 and 61 and noting that the static contribution of all modes, $(\delta_k^s)^2$, has been included in the solution because of the NASTRAN modal synthesis formulation, the expressions for kinetic and potential energy of the excluded modes become

$$\delta T_k = \frac{1}{2}\mu_k s^2(2\delta_k^s + \delta_k^d)\delta_k^d , \tag{62}$$

and

$$\delta V_k = \frac{1}{2}\mu_k\omega_k^2(2\delta_k^s + \delta_k^d)\delta_k^d . \tag{63}$$

202

4.7.4 User Input Modes

A method for using structural test data to represent a component substructure in NASTRAN modal synthesis is described in this section. The data which are required for the method and additional data that may be available and used are listed below, followed by the matrix operations used in development of the modal synthesis matrix terms for input to NASTRAN.

The following data items are used by the user mode option to modal synthesis:

1. Natural frequency, ω_j, and modal mass, μ_j, for each mode (required).
2. Modal deflections of the boundary points, $\{\phi_{ij}\}$, which were free in the modal test (required).
3. Modal reaction forces, $\{q_{bj}\}$, acting on boundary points which were fixed for the modal test (required). Note that all displacement coordinates must be classified as boundary coordinates and either deflections or forces are required for every boundary point.
4. The reduced stiffness matrix for the boundary points (optional). These data may be available from analysis or from an influence coefficient test.
5. Additional mass, not included in the modal data, on fixed boundary points (optional).

4.7.4.1 Fixed Boundary Points

Equation 27 from Section 4.7.1.3 provides the basis for this formulation. The inertia relief modes are not included and $\{u_b\}$ represents the boundary coordinates fixed in the modal test. Thus,

$$\begin{Bmatrix} u_b \\ \hdashline u_i \end{Bmatrix} = \left[\begin{array}{c:c} I & 0 \\ \hdashline G_{ib} & \phi_i \end{array}\right] \begin{Bmatrix} u_b \\ \hdashline \delta \end{Bmatrix} . \qquad (64)$$

Using Equation 64, the stiffness and mass matrices of the free (unconstrained) substructure would take the form

$$[K] = \left[\begin{array}{c:c} \bar{K}_{bb} & 0 \\ \hdashline 0 & \lceil K \rfloor \end{array}\right] , \qquad (65)$$

where $[\lceil K \rfloor]$ is diagonal and $K_j = \mu_j \omega_j^2$ for the j^th^ mode, $\bar{K}_{bb}$ is the reduced boundary stiffness matrix, and

The two coordinate systems are related as follows:

$$\left.\begin{aligned} x &= (1-\xi)(1-\eta)\, x_1 + \xi(1-\eta)\, x_2 + \xi\eta x_3 + (1-\xi)\eta\, x_4 \\ y &= (1-\xi)(1-\eta)\, y_1 + \xi(1-\eta)\, y_2 + \xi\eta y_3 + (1-\xi)\eta\, y_4 \end{aligned}\right\} , \qquad (1)$$

where the subscripts refer to grid point numbers. The displacement components along the x- and y-directions are denoted by u and v, respectively, and are assumed to vary with ξ and η in the following manner:

$$\left.\begin{aligned} u(\xi,\eta) &= (1-\xi)(1-\eta)u_1 + \xi(1-\eta)u_2 + \xi\eta u_3 + (1-\xi)\eta\, u_4 \\ v(\xi,\eta) &= (1-\xi)(1-\eta)v_1 + \xi(1-\eta)v_2 + \xi\eta v_3 + (1-\xi)\eta\, v_4 \end{aligned}\right\} . \qquad (2)$$

Properties of the assumed displacement field are that on lines of constant ξ, u and v vary linearly with η, and on lines of constant η, u and v vary linearly with ξ. In particular u and v vary linearly on the edges between grid points and as a result, displacements of adjacent elements are matched all along their common edges. Thus, the element is a "conforming" element as defined in Reference 2. It is noted from a comparison of Equations 1 and 2 that the equations which relate the displacements at any point in the element to its grid point values are identical in form to the corresponding equations for the x and y coordinates. Thus, the term "isoparametric" is used to characterize the element.

As mentioned previously, the four grid points which define the quadrilateral need not be coplanar. If they are not, a mean plane is defined as shown in Figure 2. The mean plane is located such that it is alternately H units above or below each grid point. The grid points are then projected normally onto this plane resulting in a modified but planar quadrilateral (as denoted by the primed grid point numbers). The element matrices are derived for the modified quadrilateral. These matrices are then transformed so that they are expressed in terms of displacements at the original (non-coplanar) grid points. As a result of the latter transformation, the matrices have stiffness contributions at each grid point against translations in three directions instead of two.

5.8.5.3 Strain and Stress Fields, Potential Energy

Membrane strains are related to the displacement components by the familiar relations

C-4

$$e_x = u_{,x} \qquad e_y = v_{,y} \qquad e_{xy} = u_{,y} + v_{,x} \tag{3}$$

where a comma indicates partial differentiation. Use of Equation 2 permits the strains in Equation 3 to be expressed in terms of ξ and η. Thus,

$$\left.\begin{aligned} e_x &= u_{,\xi}\,\xi_{,x} + u_{,\eta}\,\eta_{,x} \\ e_y &= v_{,\xi}\,\xi_{,y} + v_{,\eta}\,\eta_{,y} \\ e_{xy} &= u_{,\xi}\,\xi_{,y} + u_{,\eta}\,\eta_{,y} + v_{,\xi}\,\xi_{,x} + v_{,\eta}\,\eta_{,x} \end{aligned}\right\} , \tag{4}$$

where

$$\left.\begin{aligned} \xi_{,x} &= \frac{1}{J}\,y_{,\eta} & \xi_{,y} &= -\frac{1}{J}\,x_{,\eta} \\ \eta_{,x} &= -\frac{1}{J}\,y_{,\xi} & \eta_{,y} &= \frac{1}{J}\,x_{,\xi} \end{aligned}\right\} , \tag{5}$$

and

$$J = \begin{vmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{vmatrix} , \tag{6}$$

is the Jacobian of the transformation between the two element coordinate systems. For a rectangular shaped element, the x and ξ directions are identical, as are the y and η directions. For this case e_x is linear with respect to y and constant with respect to x, and e_y is linear with respect to x and constant with respect to y. The shear strain e_{xy} varies linearly with respect to both x and y. For nonrectangular-shaped elements the strain behavior is not linear with position.

The strain-displacement relations may be written in a convenient matrix form by combining Equations 1 to 5 as follows:

$$\begin{Bmatrix} e_x \\ e_y \\ e_{xy} \end{Bmatrix} = \left[A\right] \left\{u_e\right\} , \tag{7}$$

where

$$\left\{u_e\right\}^T = \left\{u_1\ v_1\ u_2\ v_2\ u_3\ v_3\ u_4\ v_4\right\}^T \quad .$$

The non-zero elements of the 3x8 matrix [A] are as follows:

$$A_{11} = (-y_4 + y_3\eta - y_{34}\xi)/J$$

$$A_{13} = (y_4 - y_4\eta + y_{34}\xi)/J$$

$$A_{15} = y_4\eta/J$$

$$A_{17} = -y_3\eta/J$$

$$A_{22} = (-x_{24} + x_{23}\eta + x_{34}\xi)/J$$

$$A_{24} = (x_{14} - x_{14}\eta - x_{34}\xi)/J$$

$$A_{26} = (x_{14}\eta - x_{12}\xi)/J$$

$$A_{28} = (-x_{12} - x_{23}\eta + x_{12}\xi)/J$$

$$A_{31} = A_{22}$$

$$A_{32} = A_{11}$$

$$A_{33} = A_{24}$$

$$A_{34} = A_{13}$$

$$A_{35} = A_{26}$$

$$A_{36} = A_{15}$$

$$A_{37} = A_{28}$$

$$A_{38} = A_{17}$$

where

$$J = -y_4x_{12} - y_{34}x_{12}\xi - (y_4x_{23} - y_3x_{14})\eta \quad , \tag{8}$$

and the components of side lengths are expressed as follows in terms of grid point coordinates:

$$x_{ij} = x_i - x_j \;, \qquad y_{ij} = y_i - y_j \quad . \tag{9}$$

The constitutive stress-strain relationships are written as

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = [G_e] \begin{Bmatrix} e_x \\ e_y \\ e_{xy} \end{Bmatrix} - [G_e] \begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_{xy} \end{Bmatrix} \bar{T} \quad , \tag{10}$$

where σ_x and σ_y are stresses in the x and y directions, respectively, σ_{xy} is the shear stress, and $[G_e]$ is a symmetric 3x3 matrix with properties referred to the element coordinate system (see p. 5.8-4). The quantities α_x, α_y, and α_{xy} are thermal expansion coefficients and $\bar{T}$ is the temperature of the element above the stress-free temperature T_o. If the element temperature is not specified directly, it is computed in terms of grid point temperatures as

$$\bar{T} = \frac{1}{4} (T_1 + T_2 + T_3 + T_4) - T_o \quad . \tag{11}$$

The potential energy for an element of thickness h including the temperature effect may be written as

$$V = \frac{h}{2} \int_0^1 \int_0^1 \begin{Bmatrix} e_x & - & \alpha_x\bar{T} \\ e_y & - & \alpha_y\bar{T} \\ e_{xy} & & \alpha_{xy}\bar{T} \end{Bmatrix}^T \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} J d\xi d\eta \quad . \tag{12}$$

Substituting Equation 10 into Equation 12 and making use of Equation 7 gives

$$V = \frac{h}{2} \{u_e\}^T \int_0^1 \int_0^1 [A]^T[G_e][A] \;\; J d\xi d\eta \; \{u_e\} - h\{u_e\}^T \int_0^1 \int_0^1 [A]^T[G_e] \begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_{xy} \end{Bmatrix} \bar{T} J d\xi d\eta \; . \tag{13}$$

The first integral represents the usual elastic strain energy of the element, and the second integral represents the thermal strain energy. An irrelevant additive constant in the above equation involving the square of known element temperature has been omitted.

5.8.5.4 Stiffness Matrix and Thermal Load Vector for the Element

The form of the potential energy written in terms of the displacement vector, $\{u_e\}$, the stiffness matrix, $[K_{ee}]$, and the thermal load vector, $\{P_e\}$, is as follows:

$$V = \frac{1}{2} \{u_e\}^T [K_{ee}] \{u_e\} - \{u_e\}^T \{P_e\} \quad . \tag{14}$$

Comparing Equations 13 and 14 yields the following formula for the required matrix and vector:

$$[K_{ee}] = h\int_0^1\int_0^1 [A]^T[G_e][A]J\,d\xi d\eta \quad . \tag{15}$$

$$\{P_e\} = h\int_0^1\int_0^1 [A]^T[G_e]\begin{Bmatrix}\alpha_x\\ \alpha_y\\ \alpha_{xy}\end{Bmatrix}\bar{T}J\,d\xi d\eta \quad . \tag{16}$$

The reader will recall that the elements of matrix [A] as well as the quantity J are functions of ξ and η. As a result, the integration indicated in Equation 15 is best performed numerically, and it will be carried out by use of Gaussian quadrature using a 4x4 grid (see Reference 3 for a discussion of the method of Gaussian quadrature). It is noted that the grid size is finer than the minimum size (2x2) required to guarantee convergence. However, preliminary studies indicated that the refined grid resulted in improved accuracy over the 2x2 grid for nonrectangular elements. The integration in Equation 16 can easily be carried out in closed form since, if the temperature, $\bar{T}$, is taken to be constant over the element, the integrand is linear in ξ and η.

The stiffness matrix and thermal load vector given in Equations 15 and 16 have been derived for an element which is assumed to be planar. If the grid points are not coplanar, then the derived element is the projection of the actual element onto the mean plane. In the latter case a transformation of the stiffness matrix and the thermal load vector is required, which relates displacements and forces at the projected grid points in the mean plane to displacements and forces at the actual grid points. It is highly desirable that the transformation produce only forces and not moments at the grid points because it is quite probable that there may be no other elements present (such as beams and bending plates) which can resist moments. Thus, the transformation can be expressed in the form

$$\{f_a\} = [B]\{f_e\} \quad , \tag{17}$$

where:

$$\{f_a\}^T = \lfloor f_{x1}, f_{y1}, f_{z1}, f_{x2}, f_{y2}, f_{z2}, f_{x3}, f_{y3}, f_{z3}, f_{x4}, f_{y4}, f_{z4} \rfloor$$

is the vector of grid point forces, and

$$\{f_e\}^T = \lfloor f_{x1}, f_{y1}, f_{x2}, f_{y2}, f_{x3}, f_{y3}, f_{x4}, f_{y4} \rfloor$$

is the vector of forces on the projected plane element.

The magnitudes of the inplane forces (f_{x1}, f_{y1}, etc.) are the same in both vectors. A method for selecting the out-of-plane forces (f_{z1}, f_{z2}, f_{z3}, f_{z4}) which satisfies the three required out-of-plane conditions of equilibrium and which exhibits symmetry with respect to permutation of grid point numbers is as follows: Let the forces on the corners of the plane quadrilateral be resolved into components colinear with the sides as shown in Figure 3a. In the edgewise view of side (a) shown in Figure 3b, the vertical force couple, f_{za}, is applied to grid points 1 and 2 so that equilibrium will be preserved when the forces f_{12} and f_{21} are transferred from the mean plane to the grid points. Thus,

$$f_{za} = \frac{H}{\ell_a}(f_{12} + f_{21}) \quad , \tag{18}$$

and in like manner, for the other three sides,

$$\left.\begin{aligned} f_{zb} &= -\frac{H}{\ell_b}(f_{23} + f_{32}) \\ f_{zc} &= \frac{H}{\ell_c}(f_{34} + f_{43}) \\ f_{zd} &= -\frac{H}{\ell_d}(f_{41} + f_{14}) \end{aligned}\right\} \quad . \tag{19}$$

The combined vertical force components at the grid points are

$$\left.\begin{aligned} f_{z1} &= f_{za} - f_{zd} \\ f_{z2} &= f_{zb} - f_{za} \\ f_{z3} &= f_{zc} - f_{zb} \\ f_{z4} &= f_{zd} - f_{zc} \end{aligned}\right\} \quad . \tag{20}$$

Generation of the elements in the rows of the [B] matrix corresponding to the vertical forces is accomplished by expressing the colinear force components, f_{12}, f_{21}, etc., in terms of the Cartesian components, f_{x1}, f_{y1}, etc., and substituting the result into Equation 20. The nonzero elements of [B] are as follows:

$$B_{11} = 1$$

$$B_{22} = 1$$

$$B_{31} = -H/\ell_a$$

$$B_{32} = -\frac{H}{\ell_d \sin\theta_1} + \frac{H\cot\theta_1}{\ell_a}$$

$$B_{33} = H/\ell_a$$

$$B_{34} = \frac{H\cot\theta_2}{\ell_a}$$

$$B_{37} = \frac{H\sin\gamma}{\ell_d\,\Delta_2}$$

$$B_{38} = \frac{H\cos\gamma}{\ell_d\,\Delta_2}$$

$$B_{43} = 1$$

$$B_{54} = 1$$

$$B_{61} = -B_{31}$$

$$B_{62} = -\frac{H\cot\theta_1}{\ell_a}$$

$$B_{63} = B_{31}$$

$$B_{64} = -\frac{H\cot\theta_2}{\ell_a} + \frac{H}{\ell_b \sin\theta_2}$$

$$B_{65} = -\frac{H\sin\gamma}{\ell_b\,\Delta_1}$$

$$B_{66} = -\frac{H\cos\gamma}{\ell_b\,\Delta_1}$$

$$B_{75} = 1$$

$$B_{86} = 1$$

$$B_{94} = -\frac{H}{\ell_b \sin\theta_2}$$

$$B_{95} = H\left(\frac{\sin\gamma}{\ell_b\,\Delta_1} + \frac{\sin\theta_2}{\ell_c\,\Delta_1}\right)$$

$$B_{96} = H\left(\frac{\cos\gamma}{\ell_b\,\Delta_1} + \frac{\cos\theta_2}{\ell_c\,\Delta_1}\right)$$

$$B_{97} = -\frac{H\sin\theta_1}{\ell_c\,\Delta_2}$$

$$B_{98} = \frac{H \cos \theta_1}{\ell_c \, \Delta_2}$$

$$B_{10,7} = 1$$

$$B_{11,8} = 1$$

$$B_{12,2} = \frac{H}{\ell_d \sin \theta_1}$$

$$B_{12,5} = - \frac{H \sin \theta_2}{\ell_c \, \Delta_1}$$

$$B_{12,6} = - \frac{H \cos \theta_2}{\ell_c \, \Delta_1}$$

$$B_{12,7} = H\left(\frac{-\sin \gamma}{\ell_d \, \Delta_2} + \frac{\sin \theta_1}{\ell_c \, \Delta_2}\right)$$

$$B_{12,8} = H\left(\frac{-\cos \gamma}{\ell_d \, \Delta_2} - \frac{\cos \theta_1}{\ell_c \, \Delta_2}\right)$$

where

$$\Delta_1 = \sin(\theta_2 - \gamma)$$

$$\Delta_2 = \sin(\theta_1 + \gamma)$$

The transformation of displacements from the mean plane to the actual grid points uses the same [B] matrix and is written

$$\{u_e\} = [B]^T \{u_a\} \quad , \tag{21}$$

where

$$\{u_a\}^T = \lfloor u_1, v_1, w_1, u_2, v_2, w_2, u_3, v_3, w_3, u_4, v_4, w_4 \rfloor$$

and w is the displacement component normal to the mean plane. In addition to the above transformation, two standard NASTRAN transformations are required. These are the element-to-basic system transformation utilizing matrix [E] and the basic-to-global system transformation utilizing the matrix [T]. Combining all three transformations results in the required global forms of the stiffness matrix $[K_{gg}]$ and thermal vector $\{P_g\}$,

$$[K_{gg}] = [T]^T[E][B][K_{ee}][B]^T[E]^T[T] \quad , \tag{22}$$

$$\{P_g\} = [T]^T[E][B]\{P_e\} \quad . \tag{23}$$

The 12x12 stiffness matrix $[K_{gg}]$ is singular with defect equal to seven. The seven defects correspond to the six rigid body motions and an unrestrained out-of-plane warping. Out-of-plane warping would also be unrestrained if the nonplanar quadrilateral were represented by a pair of triangles but not if it were represented by two pairs of overlapping triangles, as in the case of the QDMEM element.

5.8.5.5 Stress Recovery

The stresses at any point (ξ,η) in the element in terms of the displacements in the element coordinate system are obtained by combining Equations 7 and 10

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = [G_e][A]\{u_e\} - [G_e]\begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_{12} \end{Bmatrix}\bar{T} \quad , \tag{24}$$

where it will be recalled that [A] is a function of ξ and η. The stresses are evaluated at the intersection of the diagonals of the mean plane, in order to be compatible with stress calculation in the NASTRAN plate bending elements. For a parallelogram, the diagonals intersect at $\xi = \eta = 1/2$ but for more general shapes the values of ξ and η at the intersection point depend on the element dimensions. The required form of the stress recovery equation in terms of the global displacements is obtained by utilizing the three transformations described previously along with Equation 24. Thus,

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} = [G_e][A][B]^T[E]^T[T]\{u_g\} - [G_e]\begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_{xy} \end{Bmatrix}\bar{T} \quad . \tag{25}$$

REFERENCES

1. Irons, B. M., "Engineering Applications of Numerical Integration in Stiffness Methods," AIAA Journal, Vol. 4, No. 11, November 1966, pp. 2035-2037.

2. Zienkiewicz, O. C., and Cheung, Y. K., The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill Publishing Company Limited, London, 1967.

3. Przemieniecki, J. S., Theory of Matrix Structural Analysis, McGraw-Hill Book Company, Inc., New York, 1968.

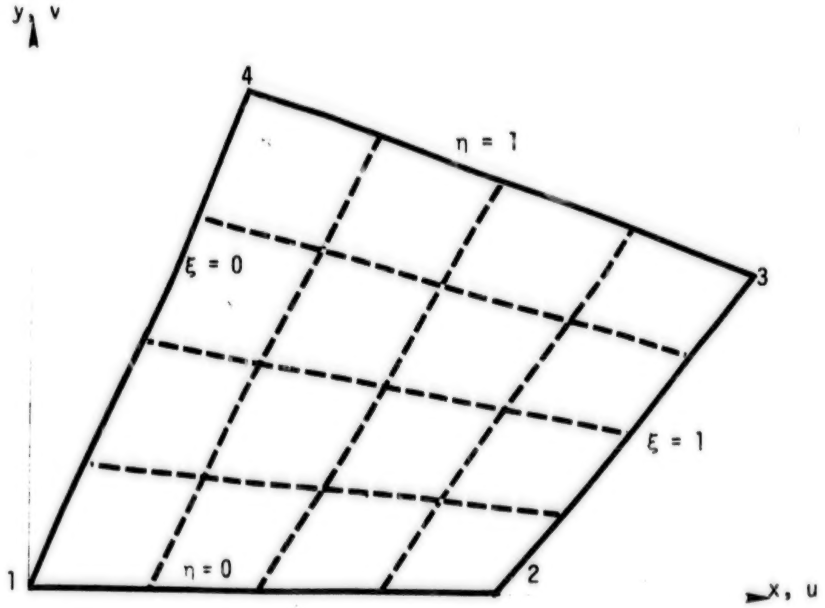

y, v
4
η = 1
ξ = 0
3
ξ = 1
1
η = 0
2
x, u

Figure 1. Coordinate systems for quadrilateral membrane element.

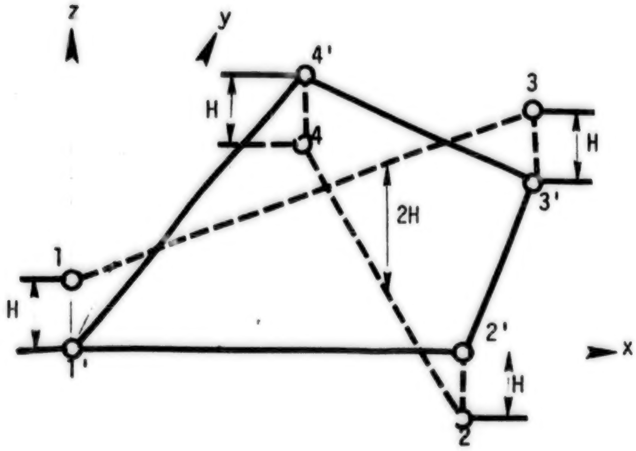
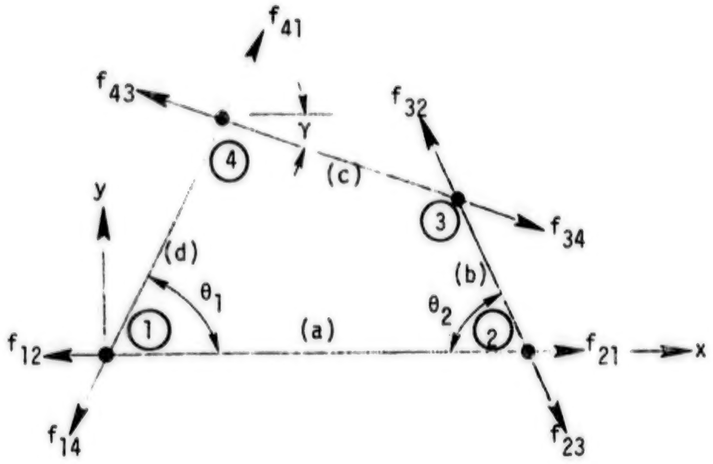
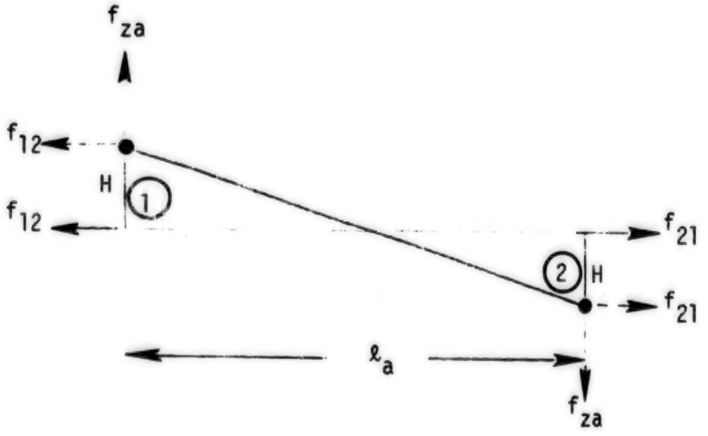

z
y
4'
H
4
3
H
3'
2H
1
H
2'
1'
x
H
2

Figure 2. Mean plane for quadrilateral membrane element.
(Actual grid points are indicated by unprimed numbers and projection
of grid points onto mean plane are indicated by primed numbers.)


f41
f43
γ
f32
4
(c)
y
3
f34
(d)
(b)
θ1
θ2
1
(a)
2
f12
f21
x
f14
f23

(a) Plan view


fza
f12
H
1
f12
f21
2
H
f21
ℓa
fza

(b) Side view

Figure 3. Method of transferring forces from a plane quadrilateral element to adjacent nonplanar grid points.

5.8.6 The Linear Strain Triangular Membrane Element, TRIM6

5.8.6.1 Introduction

This element was first formulated by J. H. Argyris and is described in References 1 and 2. The present development is based on the derivation in Reference 2. The important characteristics of the element are:

1. The stresses and strains vary linearly within the element.
2. Bilinear variation in the planar coordinates for the thickness of the element is permitted.
3. Bilinear variation in the planar coordinates for the temperature in the element is provided.

Differential stiffness and piecewise linear analysis capability are not implemented at present.

The element is compared with theoretical results for accuracy in Section 15.4. The calculation of its mass properties is discussed in section 5.8.4.

5.8.6.2 Geometry and Displacement Field

The geometry of the element is shown in Figure 1. The element has six grid points, three at the vertices and three at the mid-points of the sides. The displacement components u and v are parallel to the x and y axes of the local (element) coordinate system. The inplane displacements at the grid points of the element are represented by the vector $\{u_e\}$ where

$$\{u_e\}^T = \lfloor u_1\ v_1\ u_2\ v_2\ u_3\ v_3\ u_4\ v_4\ u_5\ v_5\ u_6\ v_6 \rfloor \tag{1}$$

Let $[K_{ee}]$ be the stiffness matrix referred to the vector $\{u_e\}$, i.e.,

$$[K_{ee}]\ \{u_e\} = \{f_e\} \quad , \tag{2}$$

where the elements of $\{f_e\}$ are the inplane forces at the grid points of the element. The stiffness matrix $[K_{ee}]$ is derived by standard finite element procedures.

The u and v displacements are assumed to vary quadratically with position on the surface of the element,

$$u = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 \tag{3}$$

PRECEDING PAGE BLANK NOT FILMED

ORIGINAL PAGE IS
OF POOR QUALITY

$$v = b_7 + b_8x + b_9y + b_{10}x^2 + b_{11}xy + b_{12}y^2 \tag{4}$$

The quantities $a_1, a_2, \ldots a_6, b_7, b_8, \ldots b_{12}$ may be regarded as generalized coordinates to which the displacements at the grid points of the element are uniquely related, i.e., the vector of generalized coordinates is expressed as

$$\{a\}^T = \lfloor a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ b_7\ b_8\ b_9\ b_{10}\ b_{11}\ b_{12} \rfloor \tag{5}$$

In concise form Equations 3 and 4 are written as

$$u = \sum_{i=1}^{6} a_i x^{m_i} y^{n_i} \tag{6}$$

$$v = \sum_{i=7}^{12} b_i x^{p_i} y^{v_i} \quad . \tag{7}$$

For convenience in later calculations, the range of summation is kept as 1 to 12 for expressions for both u and v, i.e.,

$$u = \sum_{i=1}^{12} a_i x^{m_i} y^{n_i} \tag{8}$$

$$v = \sum_{i=1}^{12} b_i x^{p_i} y^{v_i} \tag{9}$$

so that

$$a_i = m_i = n_i = 0, \quad i = 7 \text{ to } 12 \tag{10}$$

$$b_i = p_i = v_i = 0, \quad i = 1 \text{ to } 6 \quad . \tag{11}$$

In matrix notation, the vector $\{u_e\}$ is written as

$$\{u_e\} = [H]\{a\} \quad , \tag{12}$$

where the 12 x 12 [H] matrix can be obtained by substituting the coordinates of the six grid points into Equations 3 and 4. Since complete polynomial expressions are chosen for the u and v displacements, the inverse of H matrix exists. Hence {a} can be expressed as

$$\{a\} = [H]^{-1} \{u_e\} \quad . \tag{13}$$

Bilinear variation in the x and y coordinates is assumed for the thickness t of the element, i.e., the thickness t of the element at any point (x,y) within the element is given by

$$t(x,y) = c_1 + c_2 x + c_3 y \quad . \tag{14}$$

In concise form, this is written as

$$t = \sum_{k=1}^{3} c_k x^{\gamma_k} y^{s_k} \quad . \tag{15}$$

The thickness of the element at the three vertices is specified as t_1, t_3, t_5. Hence the coefficients c_1, c_2, c_3 can be expressed as

$$c_1 = \frac{t_1 a + t_3 b}{(a + b)} \tag{16}$$

$$c_2 = \frac{t_3 - t_1}{(a + b)} \tag{17}$$

$$c_3 = \frac{1}{c} (t_5 - c_1) \quad , \tag{18}$$

where a, b and c are the projected lengths of the triangle on the local x and y axes and are obtained from the basic coordinates of the vertices of the triangle as given in Section 4.87.21.2 of the Programmer's Manual.

5.8.6.3 Strain and Stress Fields

The membrane strains are

$$\varepsilon_x = \frac{\partial u}{\partial x} = a_2 + 2a_4 x + a_5 y = \sum a_i m_i x^{(m_i - 1)} y^{n_i} \tag{19}$$

$$\varepsilon_y = \frac{\partial v}{\partial y} = b_9 + b_{11}x + 2b_{12}y = \sum b_i v_i x^{p_i} y^{(v_i-1)} \tag{20}$$

$$\gamma = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \sum \left(a_i n_i x^{m_i} y^{(n_i-1)} + b_i p_i x^{(p_i-1)} y^{v_i} \right) \quad . \tag{21}$$

The stress vector $\{\sigma\}$ is related to the strain vector by the two-dimensional elastic modulus matrix, $[G_e]$

$$\{\sigma\} = [G_e] \, \{\varepsilon\} \quad . \tag{22}$$

The specification of $[G_e]$ for isotropic and anisotropic materials is the same as that given by Equations 13, 14, and 15 on page 5.8.4.

The membrane strain energy of the element is

$$E_s = \frac{1}{2} \iint \{\sigma\}^T \, \{\varepsilon\} \, t dx dy \quad . \tag{23}$$

By virtue of Equation 22 and the symmetry of matrix $[G_e]$

$$E_s = \frac{1}{2} \iint \{\varepsilon\}^T \, [G_e] \, \{\varepsilon\} \, t dx dy \quad . \tag{24}$$

Substitution of Equation 15 into Equation 24 results in

$$E_s = \frac{1}{2} \iint \{\varepsilon\}^T \, [G_e] \, \{\varepsilon\} \left(\sum_{k=1}^{3} c_k x^{\gamma_k} y^{s_k} \right) dx dy \quad . \tag{25}$$

Expressing the elements of the symmetric portion of the matrix $[G_e]$ by G_{11}, G_{12}, G_{13}, G_{22}, G_{23}, G_{33}, i.e.,

$$[G_e] = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ & G_{22} & G_{23} \\ \text{sym} & & G_{33} \end{bmatrix} \tag{26}$$

and performing the matrix multiplication of Equation 25, the expression for strain energy becomes

$$
\begin{aligned}
E_s = \frac{1}{2}\iint \Big\{ & \epsilon_x{}^2 G_{11} + \epsilon_y{}^2 G_{22} + \gamma^2 G_{33} + G_{12}\,(\epsilon_x\epsilon_y + \epsilon_y\epsilon_x) \\
& + G_{13}\,(\epsilon_x\gamma + \gamma\epsilon_x) + G_{23}\,(\epsilon_y\gamma + \gamma\epsilon_y)\Big\} \sum \left(c_k x^{\gamma_k} y^{s_k}\right) dxdy
\end{aligned}
\qquad (27)
$$

To proceed further it is necessary to have a formula for the integral of the type

$$\iint x^m y^n \, dxdy$$

taken over the area of the element. The value of the integral is found in Reference 3:

$$\iint x^m y^n dxdy = F(m,n) = c^{n+1} \left\{ a^{m+1} - (-b)^{m+1} \right\} \frac{m!n!}{(m + n + 2)!} \quad . \qquad (28)$$

Using Equations 19, 20, 21, and 28 in Equation 27, a typical term of Equation 27 becomes

$$
\begin{aligned}
& \frac{1}{2}\iint \epsilon_x{}^2 G_{11} \left(\sum c_k x^{\gamma_k} y^{s_k}\right) dxdy \\
& = \frac{1}{2} \sum_i \sum_j \sum_k \; a_i a_j c_k m_i m_j G_{11} F(m_i + m_j + \gamma_k - 2,\; n_i + n_j + s_k) \quad .
\end{aligned}
\qquad (29)
$$

Similarly, the other terms of Equation 27 can be expressed in terms of the area integral F.

5.8.6.4 Stiffness Matrix for the Element

The strain energy, E_s, can also be expressed as

$$E_s = \frac{1}{2}\,\{a\}^T\,[k_{gen}]\,\{a\} \quad , \qquad (30)$$

where $[k_{gen}]$ is the stiffness matrix with respect to generalized coordinates $\{a\}$. It can be shown that the elements of the matrix $[k_{gen}]$ are given by

$$
\begin{aligned}
k_{ij} = \sum_{k=1}^{3} c_k \Big[& G_{11} m_i m_j F(m_i + m_j + \gamma_k - 2,\; n_i + n_j + s_k) \\
& + G_{22} v_i v_j F(p_i + p_j + \gamma_k,\; v_i + v_j + s_k - 2) \\
& + G_{33} \Big\{ n_i n_j F(m_i + m_j + \gamma_k,\; n_i + n_j + s_k - 2)
\end{aligned}
$$

$$
\begin{aligned}
&\quad + p_i p_j F(p_i + p_j + \gamma_k - 2,\ v_i + v_j + s_k) \\
&\quad + n_i p_j F(m_i + p_j + \gamma_k - 1,\ n_i + v_j + s_k - 1) \\
&\quad + p_i n_j F(m_j + p_i + \gamma_k - 1,\ n_j + v_i + s_k - 1)\Big\} \\
&+ G_{12} \Big\{ m_j v_i F(m_j + p_i + \gamma_k - 1,\ n_j + v_i + s_k - 1) \\
&\quad + m_i v_j F(m_i + p_j + \gamma_k - 1,\ n_i + v_j + s_k - 1)\Big\} \\
&+ G_{13} \Big\{ (m_j n_i + m_i n_j) F(m_i + m_j + \gamma_k - 1,\ n_i + n_j + s_k - 1) \\
&\quad + m_j p_i F(m_j + p_i + \gamma_k - 2,\ n_j + v_i + s_k) \\
&\quad + m_i p_j F(m_i + p_j + \gamma_k - 2,\ n_i + v_j + s_k)\Big\} \\
&+ G_{23} \Big\{ (p_i v_j + p_j v_i) F(p_i + p_j + \gamma_k - 1,\ v_i + v_j + s_k - 1) \\
&\quad + n_i v_j F(m_i + p_j + \gamma_k,\ n_i + v_j + s_k - 2) \\
&\quad + n_j v_i F(m_j + p_i + \gamma_k,\ n_j + v_i + s_k - 2)\Big\}\Big] \quad . \qquad (31)
\end{aligned}
$$

Using Equation 13, the generalized stiffness matrix $[k_{gen}]$ can be transformed to the element stiffness matrix $[k_{ee}]$ as

$$[k_{ee}] = [H^{-1}]^T\ [k_{gen}]\ [H^{-1}] \quad . \qquad (32)$$

As a final step, the stiffness matrix is transformed from local element coordinate system to the basic coordinate system of the grid points and to the global coordinate system. Let the transformation for displacements be

$$\{u_{basic}\} = [E]^T\ \{u_{element}\} \qquad (33)$$

and

$$\{u_{global}\} = [T]\ \{u_{basic}\} \quad . \qquad (34)$$

Then,

$$[k_{basic}] = [E]\ [k_{ee}]\ [E]^T \qquad (35)$$

and

$$[k_{gg}] = [T]^T\ [k_{basic}]\ [T] \quad . \qquad (36)$$

Substituting Equation 35 in Equation 36, the global stiffness matrix becomes

$$[k_{gg}] = [T]^T\ [E]\ [k_{ee}]\ [E]^T\ [T] \quad . \qquad (37)$$

5.8.6.5 Thermal Load Vector for the Element

Thermal expansion of an element produces equivalent loads at the grid points. Thermal expansion is represented by a vector of thermal strains:

$$\{\varepsilon_t\} = \begin{Bmatrix} \varepsilon_{xt} \\ \varepsilon_{yt} \\ \gamma_t \end{Bmatrix} = \begin{Bmatrix} \alpha_{e1} \\ \alpha_{e2} \\ \alpha_{e3} \end{Bmatrix} (\overline{T} - T_o) = \{\alpha_e\} (\overline{T} - T_o) \quad , \tag{38}$$

where $\{\alpha_e\} = [U]^{-1} \{\alpha_m\}$ is a vector of thermal expansion coefficients, $[U]$ is the strain transformation matrix given in Equation 15 on page 5.8.4 and $\{\alpha_m\}$ is the vector of thermal expansion coefficients in the material axis system; T_o is the reference or stress-free temperature of the material, and $\overline{T}$ is the temperature at any point (x,y) in the element and is given by a linear polynomial

$$\overline{T} = d_1 + d_2 x + d_3 y \quad . \tag{39}$$

In concise form, this is written as

$$\overline{T} = \sum_{\ell=1}^{3} d_\ell x^{t_\ell} y^{u_\ell} \quad . \tag{40}$$

The temperatures $\overline{T}_1$, $\overline{T}_3$, and $\overline{T}_5$ at the three vertices of the element will be modified by the reference temperature T_o and used to evaluate the three constants d_1, d_2, and d_3 as

$$d_1 = \frac{T_1' a + T_3' b}{(a + b)} \tag{41}$$

$$d_2 = \frac{T_3' - T_1'}{(a + b)} \tag{42}$$

$$d_3 = \frac{1}{c} [T_5' - d_1] \quad , \tag{43}$$

where

$$T_1' = (\overline{T}_1 - T_o); \; T_3' = (\overline{T}_3 - T_o); \text{ and } T_5' = (\overline{T}_5 - T_o) \quad . \tag{44}$$

326

An equivalent elastic state of stress that will produce the same thermal strains is

$$\{\sigma_t\} = [G_e]\,\{\epsilon_t\} = [G_e]\,\{\alpha_e\}\,(\bar{T} - T_o) \quad . \tag{45}$$

An equivalent set of generalized loads $\{P_{gen}\}$ applied to grid points of the element is obtained from the relation

$$\begin{aligned}\{a\}^T\,\{P_{gen}\} &= \int_A \{\epsilon\}^T\,\{\sigma_t\}\,tdA \\ &= \iint \{\epsilon\}^T\,[G_e]\,\{\alpha_e\}\left(\sum_{\ell=1}^{3} d_\ell x^{t_\ell} y^{u_\ell}\right) \\ &\quad \left(\sum_{k=1}^{3} c_k x^{\gamma_k} y^{s_k}\right) dxdy \quad .\end{aligned} \tag{46}$$

Performing the matrix multiplications in Equation 46 and using the following notations, viz.,

$$G'_{11} = G_{11}\alpha_{e1} + G_{12}\alpha_{e2} + G_{13}\alpha_{e3} \tag{47}$$

$$G'_{22} = G_{12}\alpha_{e1} + G_{22}\alpha_{e2} + G_{23}\alpha_{e3} \tag{48}$$

$$G'_{33} = G_{13}\alpha_{e1} + G_{23}\alpha_{e2} + G_{33}\alpha_{e3} \quad , \tag{49}$$

Equation 46 reduces to

$$\begin{aligned}\{a\}^T\,\{P_{gen}\} &= \iint (\epsilon_x G'_{11} + \epsilon_y G'_{22} + \gamma G'_{33}) \\ &\left(\sum_{\ell=1}^{3} d_\ell x^{t_\ell} y^{u_\ell}\right)\left(\sum_{k=1}^{3} c_k x^{\gamma_k} y^{s_k}\right) dxdy \quad .\end{aligned} \tag{50}$$

Performing the integration term by term, the first term in Equation 50 becomes

$$\iint \epsilon_x G'_{11} \left(\sum_{\ell=1}^{3} d_\ell x^{t_\ell} y^{u_\ell}\right) \left(\sum_{k=1}^{3} c_k x^{\gamma_k} y^{s_k}\right) dxdy$$

$$= \sum_i \sum_k \sum_\ell G'_{11} a_i m_i c_k d_\ell \iint x^{(m_i + \gamma_k + t_\ell - 1)} y^{(n_i + s_k + u_\ell)} dxdy$$

$$= \sum_i \sum_k \sum_\ell G'_{11} a_i m_i c_k d_\ell F(m_i + \gamma_k + t_\ell - 1, n_i + s_k + u_\ell) \qquad (51a)$$

Similarly, the second and third terms of Equation 50 reduce to

$$\sum_i \sum_k \sum_\ell G'_{22} b_i v_i c_k d_\ell F(p_i + \gamma_k + t_\ell, v_i + s_k + u_\ell - 1) \qquad (51b)$$

and

$$\sum_i \sum_k \sum_\ell G'_{33} c_k d_\ell \{a_i n_i F(m_i + \gamma_k + t_\ell, n_i + s_k + u_\ell - 1)$$

$$+ b_i p_i F(p_i + \gamma_k + t_\ell - 1, v_i + s_k + u_\ell)\} \qquad (51c)$$

respectively. From Equation 50 and the results given by 51a, 51b, and 51c, the ith element of the generalized load vector $\{P_{gen}\}$ is

$$(P_{gen})_i = \sum_k \sum_\ell c_k d_\ell \Big[G'_{11} m_i F(m_i + \gamma_k + t_\ell - 1, n_i + s_k + u_\ell)$$

$$+ G'_{22} v_i F(p_i + \gamma_k + t_\ell, v_i + s_k + u_\ell - 1)$$

$$+ G'_{33} \{n_i F(m_i + \gamma_k + t_\ell, n_i + s_k + u_\ell - 1)$$

$$+ p_i F(p_i + \gamma_k + t_\ell - 1, v_i + s_k + u_\ell)\}\Big] \qquad (52)$$

The generalized equivalent load vector $\{P_{gen}\}$ is transformed to the load vector $\{P_e\}$ in global coordinates by the following transformations:

$$\{P_e\} = [H^{-1}]^T \{P_{gen}\} \qquad (53)$$

and

$$\{P_e\} = [T]^T[E]\{P_e\} \quad . \tag{54}$$

5.8.6.6 Stress Recovery

After the grid point displacements have been evaluated, stresses in the element are computed by combining the relationships

$$\{u_e\} = [E][T]^T\{u_g\} \tag{55}$$

$$\{a\} = [H^{-1}]\{u_e\} \quad , \tag{56}$$

and $\{\varepsilon\}$ which is evaluated from Equations 19, 20, and 21. Stress vector $\{\sigma\}$ is then equal to

$$\{\sigma\} = [G_e]\,(\{\varepsilon\} - \{\varepsilon_t\}) \quad . \tag{57}$$

The stresses are computed at the three vertices and at the centroid. The principal stresses and the maximum shear force are computed from the elements of $\{\sigma\}$. The direction of the maximum principal stress is referenced to the side joining grid points 1 and 3 of the triangle.

REFERENCES

1. Argyris, J. H., "Triangular Elements with Linearly Varying Strain for the Matrix Displacement Method," J. Royal Aero. Society, Vol. 69, October 1965, pp. 711-713.

2. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill Book Co., 1971.

3. Cowper, G. R., Kosko, E., Lindberg, G. M., and Olson, M. D., "A High Precision Triangular Plate Bending Element," Aeronautical Report LR-514, National Research Council, Ottawa, Canada, December 1968.

ORIGINAL PAGE IS
OF POOR QUALITY

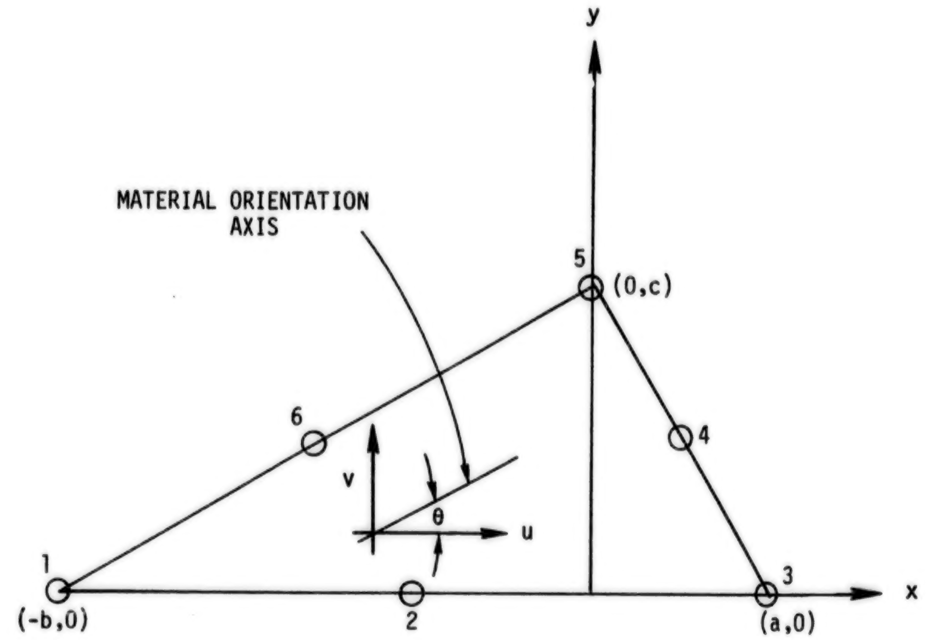

y
MATERIAL ORIENTATION AXIS
5
(0,c)
6
4
v
θ
u
1
3
x
(-b,0)
2
(a,0)

Figure 1. TRIM6 triangular membrane element in element coordinate system.

5.8.7 Higher Order Triangular Plate Bending Element, TRPLT1

5.8.7.1 Introduction

This element was developed by Narayanaswami, References 1 and 2. It is a modification of the high precision bending element of Cowper, et al., Reference 3. The element has grid points at the vertices and at the mid-points of the sides of the triangle. At each grid point, there are three degrees of freedom, viz., the transverse displacement, w, normal to the x-y plane, with positive direction outward from the paper, and the rotations, α and β, about the x and y axes, respectively. with positive directions following from the right hand rule shown in Figure 1. The element thus has 18 degrees of freedom. The transverse displacement, w, at any point within and on the boundaries of the element is assumed to vary as a quintic polynomial. Since the variation of deflection along any edge is a quintic polynomial in the edgewise coordinate, the six coefficients of this polynomial are uniquely determined by deflection and edgewise slope at the three grid points of the edge. Displacements are thus continuous between two elements that have a common edge. The rotation about each edge is constrained to vary cubically; however, since the rotations are defined only at three points along an edge, there is no rotation continuity between two elements that have a common edge. The element thus belongs to the class of nonconforming elements. The requirement that the edge rotation varies cubically along each edge establishes three constraint equations among the coefficients of the quintic polynomial for w. These equations together with the 18 relations between the grid point degrees of freedom and the polynomial coefficients serve to evaluate uniquely the coefficients a_1 through a_{21} of the quintic polynomial assumed for the transverse displacement.

5.8.7.2 Element Geometry

Rectangular cartesian coordinates are used in the formulation. An arbitrary triangular element is shown in Figure 1. X, Y, and Z are the basic coordinates; x, y, and z are the local coordinates. The grid points of the element are numbered in counter-clockwise direction as shown in the figure.

The following relationships between the X and x axes, and the coordinates of the vertices of the element, can be easily derived from Figure 1.

$$\cos\theta = \frac{X_3 - X_1}{\gamma} \quad , \qquad \sin\theta = \frac{Y_3 - Y_1}{\gamma} \tag{1}$$

$$a = (X_3 - X_5)\cos\theta - (Y_5 - Y_3)\sin\theta$$

$$= \frac{(X_3 - X_5)(X_3 - X_1) + (Y_3 - Y_5)(Y_3 - Y_1)}{\gamma} \tag{2}$$

$$b = \frac{(X_5 - X_1)(X_3 - X_1) + (Y_5 - Y_1)(Y_3 - Y_1)}{\gamma} \tag{3}$$

$$c = \frac{(X_3 - X_1)(Y_5 - Y_1) - (Y_3 - Y_1)(X_5 - X_1)}{\gamma} \tag{4}$$

where

$$\gamma = [(X_3 - X_1)^2 + (Y_3 - Y_1)^2]^{1/2} \tag{5}$$

5.8.7.3 Displacement Field

The deflection w(x,y) within the triangular element is assumed to vary as a quintic polynomial in the local coordinates, that is,

$$\begin{aligned} w(x,y) = {} & a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 + a_7x^3 \\ & + a_8x^2y + a_9xy^2 + a_{10}y^3 + a_{11}x^4 + a_{12}x^3y \\ & + a_{13}x^2y^2 + a_{14}xy^3 + a_{15}y^4 + a_{16}x^5 + a_{17}x^4y \\ & + a_{18}x^3y^2 + a_{19}x^2y^3 + a_{20}xy^4 + a_{21}y^5 \quad . \end{aligned} \tag{6}$$

In concise form, this is written as

$$w = \sum_{i=1}^{21} a_i x^{m_i} y^{n_i} \tag{6a}$$

There are 21 independent coefficients, a_1 through a_{21}. These are evaluated by the following procedure.

333

The element has 18 degrees of freedom; namely, lateral displacement w in the z-direction, rotation α about the x-axis, and rotation β about the y-axis at each of the six grid points. The rotations α and β are obtained from the definitions of transverse shear strains γ_{xz} and γ_{yz}, that is

$$\gamma_{xz} = \frac{\partial w}{\partial x} + \beta \quad \text{and} \quad \gamma_{yz} = \frac{\partial w}{\partial y} - \alpha \quad . \tag{7}$$

It is shown later on that γ_{xz} and γ_{yz}, and hence α and β, at any grid point can be expressed in terms of the coefficients a_1 through a_{21}. Thus, 18 equations relating w, α and β at the grid points to the 21 coefficients are obtained. Three additional relations are required so that the 21 coefficients can be uniquely determined. These relations are obtained by imposing the condition that the edge rotation varies cubically along each edge. It is clear that these three constraint equations involve only the coefficients of the fifth degree terms in Equation 6, since the lower degree terms satisfy the condition of cubic edge rotation automatically. Moreover, the condition depends only on the orientation of an edge. Along the edge defined by grid points 1 and 3 (where $y = 0$), the condition of the cubic edge rotation requires that

$$a_{17} = 0 \quad . \tag{8}$$

Along the edge defined by grid points 1 and 5 (inclined at angle δ to the x-axis), the edge rotation r_e is given by

$$\begin{aligned} r_e = \beta \sin \delta + \alpha \cos \delta = -(&5a_{16}x^4 + 4a_{17}x^3y + 3a_{18}x^2y^2 \\ &+ 2a_{19}xy^3 + a_{20}y^4) \sin \delta + (a_{17}x^4 + 2a_{18}x^3y + 3a_{19}x^2y^2 \\ &+ 4a_{20}xy^3 + 5a_{21}y^4) \cos \delta + . . . \end{aligned} \tag{9}$$

where the dots indicate terms of third or lower degree. Also, along this edge,

$$x = s \cos \delta \quad \text{and} \quad y = s \sin \delta \quad , \tag{10}$$

where s is the distance along the edge and

$$\cos \delta = b/\sqrt{b^2 + c^2} \quad \text{and} \quad \sin \delta = c/\sqrt{b^2 + c^2} \quad . \tag{11}$$

By substituting x and y from Equation 10 and cos δ and sin δ from Equations 11 into Equation 9 and rearranging (so that the leading terms are positive), the condition for cubic variation of rotation about edge 1-5 is

$$5b^4ca_{16} + (4b^3c^2 - b^5)a_{17} + (3b^2c^3 - 2b^4c)a_{18} + (2bc^4 - 3b^3c^2)a_{19} + (c^5 - 4b^2c^3)a_{20} - 5bc^4a_{21} = 0 \quad . \qquad (12)$$

Similarly, the condition for cubic variation of the rotation about the edge defined by grid points 3 and 5 (Figure 1) can be written as

$$5a^4ca_{16} + (-4a^3c^2 + a^5)a_{17} + (3a^2c^3 - 2a^4c)a_{18} + (-2ac^4 + 3a^3c^2)a_{19} + (c^5 - 4a^2c^3)a_{20} + 5ac^4a_{21} = 0 \quad . \qquad (13)$$

The 18 relations between grid point displacements and the coefficients of the polynomial in Equation 6 are written as

$$\{\delta\} = [Q]\,\{a\} \quad , \qquad (14)$$

where $\{\delta\}$ is the vector of grid points displacements, [Q] is the (18 x 21) matrix involving the coordinates of grid points substituted into the functions w, Equation 6, and the appropriate expressions of α and β derived in detail later, and {a} is the column vector of coefficients a_1 through a_{21}. The [Q] matrix is now augmented by the three constraint Equations 8, 12, and 13 to form a new (21 x 21) matrix [R] in the following equation:

$$\{\delta_a\} = [R]\,\{a\} \quad , \qquad (15)$$

where

$$\{\delta_a\} = \begin{Bmatrix} \{\delta\} \\ 0 \\ 0 \\ 0 \end{Bmatrix} \quad . \qquad (15a)$$

For use in the evaluation of the stiffness matrix, {a} needs to be expressed in terms of $\{\delta_a\}$ and, hence, it has to be established that the inverse of matrix [R] exists. The non-singularity of such a matrix [R] for the T-15 and T-21 elements of Bell (Reference 4) follows from the completeness of the polynomials for w. For the high precision element, Cowper et al. (Reference 3) give an explicit expression for the determinant of such a matrix and show that the matrix is non-singular in all practical situations. For this element, a numerical experiment described in Reference 1 verifies that R is non-singular for all practical cases. Hence Equation 15 is inverted to give

$$\{a\} = [R]^{-1}\,\{\delta_a\} \quad . \tag{16}$$

This equation can also be written as

$$\{a\} = [S]\,\{\delta\} \quad , \tag{17}$$

where [S] is a (21 x 18) matrix and consists of the first 18 columns of $[R]^{-1}$.

From the computational standpoint, it is advantageous to substitute Equation 8 into Equation 6 and replace coefficients a_{18} through a_{21} by coefficients a_{17} through a_{20}, respectively. The matrix [Q] then is of size (18 x 20); [R] becomes (20 x 20), and [S] becomes (20 x 18). To add to the clarity of presentation, however, the complete quintic polynomial for w in Equation 6 is retained throughout this section and matrices [Q], [R], and [S] and vector {a} will have sizes (18 x 21), (21 x 21), (21 x 18), and (21 x 1), respectively.

5.8.7.4 Elastic Relationships

The elastic relationships are obtained from the theory of deformation for plates (Reference 5). The curvatures are defined by

$$\begin{Bmatrix} \chi_x \\ \chi_y \\ \chi_{xy} \end{Bmatrix} = \begin{Bmatrix} -\dfrac{\partial\beta}{\partial x} \\ \dfrac{\partial\alpha}{\partial y} \\ \dfrac{\partial\alpha}{\partial x} - \dfrac{\partial\beta}{\partial y} \end{Bmatrix} \quad . \tag{18}$$

Bending and twisting moments are related to curvatures by

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = [D] \begin{Bmatrix} \chi_x \\ \chi_y \\ \chi_{xy} \end{Bmatrix} \quad , \tag{19}$$

where [D] is, in general, a full symmetric matrix of elastic coefficients. For a solid isotropic plate of uniform thickness t,

$$[D] = \frac{Et^3}{12(1 - \nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix} . \tag{20}$$

The thickness t of the element is assumed to vary bilinearly with position over the surface,

$$t = c_1 + c_2x + c_3y \quad . \tag{21}$$

In concise form, it is written as

$$t = \sum_{k=1}^{3} c_k x^{\gamma_k} y^{s_k} \quad . \tag{21a}$$

The thickness of the three vertices of the element t_1, t_3, and t_5 will be used to evaluate the constants c_1, c_2, and c_3. It can be shown that

$$c_1 = \frac{t_1a + t_3b}{(a + b)} \tag{22}$$

$$c_2 = \frac{t_3 - t_1}{(a + b)} \tag{23}$$

and

$$c_3 = \frac{1}{c}(t_5 - c_1) \quad . \tag{24}$$

For an isotropic plate, [D] becomes

$$[D] = \frac{1}{12}[G_e]\left(\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} c_ic_jc_k x^{(\gamma_i+\gamma_j+\gamma_k)} y^{(s_i+s_j+s_k)}\right) \quad , \tag{25}$$

where

$$[G_e] = \begin{bmatrix} \frac{E}{1 - \nu^2} & \frac{E\nu}{1 - \nu^2} & 0 \\ \frac{E\nu}{1 - \nu^2} & \frac{E}{1 - \nu^2} & 0 \\ 0 & 0 & \frac{E}{2(1 + \nu)} \end{bmatrix} . \tag{26}$$

337

For anisotropic materials with the material orientation axis inclined at ϕ to the x-axis, the material elastic modulus matrix $[D_m]$ is transformed to the element elastic modulus matrix by

$$[D] = [U]^T \, [D_m] \, [U] \quad , \tag{27}$$

where

$$[U] = \begin{bmatrix} \cos^2\phi & \sin^2\phi & \cos\phi\,\sin\phi \\ \sin^2\phi & \cos^2\phi & -\cos\phi\,\sin\phi \\ -2\cos\phi\,\sin\phi & 2\cos\phi\,\sin\phi & \cos^2\phi - \sin^2\phi \end{bmatrix} \quad . \tag{28}$$

The positive sense of bending and twisting moments and transverse shear resultants is shown in Figure 2.

The moment equilibrium equations are written as

$$V_x + \frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y} = 0 \tag{29}$$

and

$$V_y + \frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x} = 0 \quad . \tag{30}$$

Transverse shear strains are related to the shear resultants by

$$\{\gamma\} = \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = [J] \begin{Bmatrix} V_x \\ V_y \end{Bmatrix} \quad . \tag{31}$$

The matrix [J] is, in general, a full (2 x 2) symmetric matrix of elements J_{11}, J_{12} ($J_{21} = J_{12}$) and J_{22}. For a plate with isotropic transverse shear material,

$$[J] = \frac{1}{Gt^*} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \tag{32}$$

where G is the shear modulus and t* is an "effective" thickness for transverse shear. For the simple case of a plate of uniform thickness t, t* has the value t.

From Equations 29, 30, and 31, if follows that

$$\left.\begin{aligned} \gamma_{xz} &= -J_{11}\left[\frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y}\right] - J_{12}\left[\frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x}\right] \\ \text{and} \qquad \gamma_{yz} &= -J_{12}\left[\frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y}\right] - J_{22}\left[\frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x}\right] \end{aligned}\right\} . \tag{33}$$

Partial differentiation with respect to x and y of Equation 19 (with subscripts on D denoting the elements of [D]) gives

$$\left.\begin{aligned} \frac{\partial M_x}{\partial x} &= D_{11}\frac{\partial \chi_x}{\partial x} + D_{12}\frac{\partial \chi_y}{\partial x} + D_{13}\frac{\partial \chi_{xy}}{\partial x} \\ \frac{\partial M_y}{\partial y} &= D_{12}\frac{\partial \chi_x}{\partial y} + D_{22}\frac{\partial \chi_y}{\partial y} + D_{23}\frac{\partial \chi_{xy}}{\partial y} \\ \frac{\partial M_{xy}}{\partial x} &= D_{13}\frac{\partial \chi_x}{\partial x} + D_{23}\frac{\partial \chi_y}{\partial x} + D_{33}\frac{\partial \chi_{xy}}{\partial x} \\ \text{and} \qquad \frac{\partial M_{xy}}{\partial y} &= D_{13}\frac{\partial \chi_x}{\partial y} + D_{23}\frac{\partial \chi_y}{\partial y} + D_{33}\frac{\partial \chi_{xy}}{\partial y} \end{aligned}\right\} , \tag{34}$$

where the symmetry of the [D] matrix has been used. By substituting Equations 34 into Equations 33,

$$\begin{aligned} \gamma_{xz} = -\, J_{11} &\left[D_{11}\frac{\partial \chi_x}{\partial x} + D_{12}\frac{\partial \chi_y}{\partial x} + D_{13}\frac{\partial \chi_{xy}}{\partial x}\right. \\ &\left. + D_{13}\frac{\partial \chi_x}{\partial y} + D_{23}\frac{\partial \chi_y}{\partial y} + D_{33}\frac{\partial \chi_{xy}}{\partial y}\right] \\ -\, J_{12} &\left[D_{12}\frac{\partial \chi_x}{\partial y} + D_{22}\frac{\partial \chi_y}{\partial y} + D_{23}\frac{\partial \chi_{xy}}{\partial y}\right. \\ &\left. + D_{13}\frac{\partial \chi_x}{\partial x} + D_{23}\frac{\partial \chi_y}{\partial x} + D_{33}\frac{\partial \chi_{xy}}{\partial x}\right] \end{aligned} \tag{35}$$

$$\begin{aligned}\gamma_{yz} = - J_{12} \Bigg[& D_{11}\frac{\partial \chi_x}{\partial x} + D_{12}\frac{\partial \chi_y}{\partial x} + D_{13}\frac{\partial \chi_{xy}}{\partial x} \\ & + D_{13}\frac{\partial \chi_x}{\partial y} + D_{23}\frac{\partial \chi_y}{\partial y} + D_{33}\frac{\partial \chi_{xy}}{\partial y} \Bigg] \\ - J_{22} \Bigg[& D_{12}\frac{\partial \chi_x}{\partial y} + D_{22}\frac{\partial \chi_y}{\partial y} + D_{23}\frac{\partial \chi_{xy}}{\partial y} \\ & + D_{13}\frac{\partial \chi_x}{\partial x} + D_{23}\frac{\partial \chi_y}{\partial x} + D_{33}\frac{\partial \chi_{xy}}{\partial x} \Bigg] \ . \end{aligned} \qquad (36)$$

Rearranging and writing Equations 29 and 30 in matrix notation yields

$$\begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} \end{bmatrix} \begin{Bmatrix} \chi_{x,x} \\ \chi_{y,x} \\ \chi_{xy,x} \\ \chi_{x,y} \\ \chi_{y,y} \\ \chi_{xy,y} \end{Bmatrix} \ , \qquad (37)$$

where a comma in the subscript denotes partial differentiation and where

$$A_{11} = -(J_{11}D_{11} + J_{12}D_{13}) \qquad (38a)$$

$$A_{12} = -(J_{11}D_{12} + J_{12}D_{23}) \qquad (38b)$$

$$A_{13} = -(J_{11}D_{13} + J_{12}D_{33}) \qquad (38c)$$

$$A_{14} = -(J_{11}D_{13} + J_{12}D_{12}) \qquad (38d)$$

$$A_{15} = -(J_{11}D_{23} + J_{12}D_{22}) \qquad (38e)$$

$$A_{16} = -(J_{11}D_{33} + J_{12}D_{23}) \qquad (38f)$$

$$A_{21} = -(J_{12}D_{11} + J_{22}D_{13}) \qquad (38g)$$

$$A_{22} = -(J_{12}D_{12} + J_{22}D_{23}) \qquad (38h)$$

$$A_{23} = -(J_{12}D_{13} + J_{22}D_{33}) \qquad (38i)$$

$$A_{24} = -(J_{12}D_{13} + J_{22}D_{12}) \qquad (38j)$$

$$A_{25} = -(J_{12}D_{23} + J_{22}D_{22}) \qquad (38k)$$

and

$$A_{26} = -(J_{12}D_{33} + J_{22}D_{23}) \qquad (38l)$$

From Equations 7 and 18, it follows that

$$\left.\begin{aligned} \chi_x &= -\frac{\partial\beta}{\partial x} = \frac{\partial^2 w}{\partial x^2} - \frac{\partial\gamma_{xz}}{\partial x} \\ \chi_y &= \frac{\partial\alpha}{\partial y} = \frac{\partial^2 w}{\partial y^2} - \frac{\partial\gamma_{yz}}{\partial y} \\ \text{and} \qquad \chi_{xy} &= \frac{\partial\alpha}{\partial x} - \frac{\partial\beta}{\partial y} = 2\,\frac{\partial^2 w}{\partial x\partial y} - \frac{\partial\gamma_{xz}}{\partial y} - \frac{\partial\gamma_{yz}}{\partial x} \end{aligned}\right\} \quad . \tag{39}$$

Shear forces (and hence shear strains) are proportional to the third derivatives of the displacements. Since the displacement within the element is assumed to vary as a quintic polynomial, shear strains are expressed by a quadratic polynomial as follows:

$$\gamma_{xz} = b_1 + b_2x + b_3y + b_4x^2 + b_5xy + b_6y^2 \tag{40}$$

and

$$\gamma_{yz} = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 \quad . \tag{41}$$

The task now is to express the unknown coefficients b_1 through b_6 and c_1 through c_6 in terms of the generalized coordinates a_1 through a_{21}. Differentiating χ_x, χ_y, and χ_{xy} and substituting w, γ_{xz}, and γ_{yz} from Equations 6, 40, and 41 into Equations 39,

$$\chi_{x,x} = \frac{\partial^3 w}{\partial x^3} - \frac{\partial^2\gamma_{xz}}{\partial x^2} = 6a_7 + 24a_{11}x + 6a_{12}y + 60a_{16}x^2 + 24a_{17}xy + 6a_{18}y^2 - 2b_4 \tag{42}$$

$$\chi_{y,x} = \frac{\partial^3 w}{\partial x\partial y^2} - \frac{\partial^2\gamma_{yz}}{\partial x\partial y} = 2a_9 + 4a_{13}x + 6a_{14}y + 6a_{18}x^2 + 12a_{19}xy + 12a_{20}y^2 - c_5 \tag{43}$$

$$\chi_{xy,x} = 2\,\frac{\partial^3 w}{\partial x^2\partial y} - \frac{\partial^2\gamma_{xz}}{\partial x\partial y} - \frac{\partial^2\gamma_{yz}}{\partial x^2} = 4a_8 + 12a_{12}x + 8a_{13}y + 24a_{17}x^2 + 24a_{18}xy + 12a_{19}y^2 - b_5 - 2c_4 \tag{44}$$

$$\chi_{x,y} = \frac{\partial^3 w}{\partial x^2 \partial y} - \frac{\partial^2 \gamma_{xz}}{\partial x \partial y} = 2a_8 + 6a_{12}x + 4a_{13}y + 12a_{17}x^2$$

$$+ 12a_{18}xy + 6a_{19}y^2 - b_5 \qquad (45)$$

$$\chi_{y,y} = \frac{\partial^3 w}{\partial y^3} - \frac{\partial^2 \gamma_{yz}}{\partial y^2} = 6a_{10} + 6a_{14}x + 24a_{15}y + 6a_{19}x^2$$

$$+ 24a_{20}xy + 60a_{21}y^2 - 2c_6 \qquad (46)$$

and

$$\chi_{xy,y} = 2\frac{\partial^3 w}{\partial x \partial y^2} - \frac{\partial^2 \gamma_{xz}}{\partial y^2} - \frac{\partial^2 \gamma_{yz}}{\partial x \partial y} = 4a_9 + 8a_{13}x + 12a_{14}y$$

$$+ 12a_{18}x^2 + 24a_{19}xy + 24a_{20}y^2 - 2b_6 - c_5 \quad . \qquad (47)$$

By substituting Equations 40 through 47 into Equations 37, the following are obtained:

$$b_1 + b_2x + b_3y + b_4x^2 + b_5xy + b_6y^2$$

$$= A_{11}(6a_7 + 24a_{11}x + 6a_{12}y + 60a_{16}x^2 + 24a_{17}xy$$

$$+ 6a_{18}y^2 - 2b_4) + A_{12}(2a_9 + 4a_{13}x + 6a_{14}y + 6a_{18}x^2$$

$$+ 12a_{19}xy + 12a_{20}y^2 - c_5) + A_{13}(4a_8 + 12a_{12}x + 8a_{13}y$$

$$+ 24a_{17}x^2 + 24a_{18}xy + 12a_{19}y^2 - b_5 - 2c_4) + A_{14}(2a_8$$

$$+ 6a_{12}x + 4a_{13}y + 12a_{17}x^2 + 12a_{18}xy + 6a_{19}y^2 - b_5)$$

$$+ A_{15}(6a_{10} + 6a_{14}x + 24a_{15}x + 6a_{19}x^2 + 24a_{20}xy$$

$$+ 60a_{21}y^2 - 2c_6) + A_{16}(4a_9 + 8a_{13}x + 12a_{14}y$$

$$+ 12a_{18}x^2 + 24a_{19}xy + 24a_{20}y^2 - 2b_6 - c_5) \quad . \qquad (48)$$

$$
\begin{aligned}
&c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6x^2 \\
&\quad = A_{21}(6a_7 + 24a_{11}x + 6a_{12}y + 60a_{16}x^2 + 24a_{17}xy \\
&\quad + 6a_{18}y^2 - 2b_4) + A_{22}(2a_9 + 4a_{13}x + 6a_{14}y + 6a_{18}x^2 \\
&\quad + 12a_{19}xy + 12a_{20}y^2 - c_5) + A_{23}(4a_8 + 12a_{12}x + 8a_{13}y \\
&\quad + 24a_{17}x^2 + 24a_{18}xy + 12a_{19}y^2 - b_5 - 2c_4) + A_{24}(2a_8 \\
&\quad + 6a_{12}x + 4a_{13}y + 12a_{17}x^2 + 12a_{18}xy + 6a_{19}y^2 - b_5) \\
&\quad + A_{25}(6a_{10} + 6a_{14}x + 24a_{15}y + 6a_{19}x^2 + 24a_{20}xy \\
&\quad + 60a_{21}y^2 - 2c_6) + A_{26}(4a_9 + 8a_{13}x + 12a_{14}y + 12a_{18}x^2 \\
&\quad + 24a_{19}xy + 24a_{20}y^2 - 2b_6 - c_5) \quad . \qquad (49)
\end{aligned}
$$

By comparing coefficients of like powers in x, y, x^2, xy, and y^2 and constants of Equations 48 and 49, the coefficients b_1 through b_6 and c_1 through c_6 can be expressed in terms of the generalized coordinates a_1 through a_{21}. Thus

$$
\left.
\begin{aligned}
b_2 &= 24A_{11}a_{11} + 6(A_{14} + 2A_{13})a_{12} + 4(A_{12} + 2A_{16})a_{13} + 6A_{15}a_{14} \\
b_3 &= 6A_{11}a_{12} + 4(A_{14} + 2A_{13})a_{13} + 6(A_{12} + 2A_{16})a_{14} + 24A_{15}a_{15} \\
b_4 &= 60A_{11}a_{16} + 12(A_{14} + 2A_{13})a_{17} + 6(A_{12} + 2A_{16})a_{18} + 6A_{15}a_{19} \\
b_5 &= 24A_{11}a_{17} + 12(A_{14} + 2A_{13})a_{18} + 12(A_{12} + 2A_{16})a_{19} + 24A_{15}a_{20} \\
b_6 &= 6A_{11}a_{18} + 6(A_{14} + 2A_{13})a_{19} + 12(A_{12} + 2A_{16})a_{20} + 60A_{15}a_{21} \\
b_1 &= 6A_{11}a_7 + 2(A_{14} + 2A_{13})a_8 + 2(A_{12} + 2A_{16})a_9 + 6A_{15}a_{10} \\
&\quad - 2A_{11}b_4 - (A_{13} + A_{14})b_5 - 2A_{16}b_6 - 2A_{13}c_4 - (A_{12} + A_{16})c_5 \\
&\quad - 2A_{15}c_6
\end{aligned}
\right\} \qquad (50)
$$

and

$$\left.\begin{aligned}
c_2 &= 24A_{21}a_{11} + 6(A_{24} + 2A_{23})a_{12} + 4(A_{22} + 2A_{26})a_{13} + 6A_{25}a_{14} \\
c_3 &= 6A_{21}a_{12} + 4(A_{24} + 2A_{23})a_{13} + 6(A_{22} + 2A_{26})a_{14} + 24A_{25}a_{15} \\
c_4 &= 60A_{21}a_{16} + 12(A_{24} + 2A_{23})a_{17} + 6(A_{22} + 2A_{26})a_{18} + 6A_{25}a_{19} \\
c_5 &= 24A_{21}a_{17} + 12(A_{24} + 2A_{23})a_{18} + 12(A_{22} + 2A_{26})a_{19} + 24A_{25}a_{20} \\
c_6 &= 6A_{21}a_{18} + 6(A_{24} + 2A_{23})a_{19} + 12(A_{22} + 2A_{26})a_{20} + 60A_{25}a_{21} \\
c_1 &= 6A_{21}a_7 + 2(A_{24} + 2A_{23})a_8 + 2(A_{22} + 2A_{26})a_9 + 6A_{25}a_{10} \\
&\quad - 2A_{21}b_4 - (A_{23} + A_{24})b_5 - 2A_{26}b_6 - 2A_{23}c_4 - (A_{22} + A_{26})c_5 \\
&\quad - 2A_{25}c_6 \quad .
\end{aligned}\right\} \tag{51}$$

If Equations 50 and 51 are substituted into Equations 40 and 41, the explicit relation between the transverse shear strain and the generalized coordinates (i.e., coefficients of the displacement polynomial) can be obtained in matrix notation as

$$\{\gamma\} = [B_1]\ \{a\} \tag{52}$$

where $[B_1]$ is a (2 x 21) matrix whose nonzero elements are as follows:

$$B_1(1,7) = 6A_{11} \tag{52a}$$

$$B_1(1,8) = 2A_{31} \tag{52b}$$

$$B_1(1,9) = 2A_{32} \tag{52c}$$

$$B_1(1,10) = 6A_{15} \tag{52d}$$

$$B_1(1,11) = 24A_{11}x \tag{52e}$$

$$B_1(1,12) = 6(A_{31}x + A_{11}y) \tag{52f}$$

$$B_1(1,13) = 4(A_{32}x + A_{31}y) \tag{52g}$$

$$B_1(1,14) = 6(A_{15}x + A_{32}y) \tag{52h}$$

$$B_1(1,15) = 24A_{15}y \tag{52i}$$

$$B_1(1,16) = -120(A^2_{11} + A_{13}A_{21} - 0.5A_{11}x^2) \tag{52j}$$

$$\begin{aligned} B_1(1,17) = {} & -24[A_{11}(A_{31} + A_{38}) + A_{13}A_{33} + A_{21}A_{39} \\ & -0.5A_{31}x^2 - A_{11}xy] \end{aligned} \tag{52k}$$

$$\begin{aligned} B_1(1,18) = {} & -12(A_{11}A_{32} + A_{13}A_{34} + A_{38}A_{31} + A_{39}A_{33} + A_{11}A_{16} \\ & + A_{15}A_{21} - 0.5A_{32}x^2 - A_{31}xy - 0.5A_{11}y^2) \end{aligned} \tag{52ℓ}$$

$$\begin{aligned} B_1(1,19) = {} & -12(A_{11}A_{15} + A_{13}A_{25} + A_{38}A_{32} + A_{39}A_{34} + A_{16}A_{31} \\ & + A_{15}A_{33} - 0.5A_{15}x^2 - A_{32}xy - 0.5A_{31}y^2) \end{aligned} \tag{52m}$$

$$\begin{aligned} B_1(1,20) = {} & -24(A_{15}A_{38} + A_{25}A_{39} + A_{16}A_{32} + A_{15}A_{34} \\ & - A_{15}xy - 0.5A_{32}y^2) \end{aligned} \tag{52n}$$

$$B_1(1,21) = -120(A_{15}A_{16} + A_{15}A_{25} - 0.5A_{15}y^2) \tag{52o}$$

$$B_1(2,7) = 6A_{21} \tag{52p}$$

$$B_1(2,8) = 2A_{33} \tag{52q}$$

$$B_1(2,9) = 2A_{34} \tag{52r}$$

$$B_1(2,10) = 6A_{25} \tag{52s}$$

$$B_1(2,11) = 24A_{21}x \tag{52t}$$

$$B_1(2,12) = 6(A_{33}x + A_{21}y) \tag{52u}$$

$$B_1(2,13) = 4(A_{34}x + A_{33}y) \tag{52v}$$

$$B_1(2,14) = 6(A_{25}x + A_{34}y) \tag{52w}$$

$$B_1(2,15) = 24A_{25}y \tag{52x}$$

$$B_1(2,16) = -120(A_{11}A_{21} + A_{23}A_{21} - 0.5A_{21}x^2) \tag{52y}$$

$$\begin{aligned} B_1(2,17) = {} & -24(A_{21}A_{31} + A_{11}A_{40} + A_{23}A_{33} + A_{21}A_{34} \\ & - 0.5A_{33}x^2 - A_{21}xy) \end{aligned} \tag{52z}$$

$$\begin{aligned} B_1(2,18) = {} & -12(A_{21}A_{32} + A_{23}A_{24} + A_{40}A_{31} + A_{41}A_{33} + A_{26}A_{11} \\ & + A_{25}A_{21} - 0.5A_{34}x^2 - A_{33}xy - 0.5A_{21}y^2) \end{aligned} \tag{52aa}$$

$$B_1(2,19) = -12(A_{21}A_{15} + A_{23}A_{25} + A_{40}A_{32} + A_{41}A_{34} + A_{26}A_{31} + A_{25}A_{33} - 0.5A_{25}x^2 - A_{34}xy - 0.5A_{33}y^2) \tag{52bb}$$

$$B_1(2,20) = -24(A_{15}A_{40} + A_{25}A_{41} + A_{26}A_{32} + A_{25}A_{34} - A_{25}xy - 0.5A_{34}y^2) \tag{52cc}$$

and
$$B_1(2,21) = -120(A_{15}A_{26} + A^2_{25} - 0.5A_{25}y^2) \quad , \tag{52dd}$$

where A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}, A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, and A_{26} are as defined in Equations 38 and

$$\left.\begin{aligned}
A_{31} &= A_{14} + 2A_{13} \\
A_{32} &= A_{12} + 2A_{16} \\
A_{33} &= A_{24} + 2A_{23} \\
A_{34} &= A_{22} + 2A_{26} \\
A_{35} &= A_{33} + A_{11} \\
A_{36} &= A_{34} + A_{31} \\
A_{37} &= A_{25} + A_{32} \\
A_{38} &= A_{13} + A_{14} \\
A_{39} &= A_{12} + A_{16} \\
A_{40} &= A_{23} + A_{24} \\
\text{and} \qquad A_{41} &= A_{22} + A_{26}
\end{aligned}\right\} \tag{53}$$

If the plate is assumed to be rigid in transverse shear, the coefficients A_{11} through A_{16} and A_{21} through A_{26} of Equations 38 are zero (since $G = \infty$) and hence coefficients b_1 through b_6 and c_1 through c_6 of Equations 40 and 41 are zero. Moreover, the transverse shear strains vary linearly with G^{-1} with $\{\gamma\}$ approaching 0 as $G \to \infty$; that is, convergence to the limiting case of zero transverse shear is uniform.

5.8.7.5 Stiffness Matrix

The strain energy for a plate may be written as

$$U = \frac{1}{2}\iint (\{M\}^T \{\chi\} + \{V\}^T \{\gamma\})\, dxdy \quad , \tag{54}$$

where {M} is the vector of bending and twisting moments per unit length, $\{\chi\}$ is the vector of curvatures, {V} is the vector of transverse shear forces per unit length, and $\{\gamma\}$ is the vector of transverse shear strains. Substituting Equations 19 and 31 into Equation 54, and using the symmetry of [D] and [J] matrices, yields

$$U = \frac{1}{2} \iint \{\chi\}^T \, [D] \, \{\chi\} + \{\gamma\}^T \, [G] \, \{\gamma\} \, dxdy \quad , \tag{55}$$

where

$$[G] = [J]^{-1} . \tag{56}$$

With $[K_{gen}]$ denoting the generalized stiffness matrix, that is, the stiffness matrix with respect to generalized coordinates (coefficients of the displacement polynomial) {a}, the strain energy can also be expressed as

$$U = \frac{1}{2} \, \{a\}^T \, [K_{gen}] \, \{a\} \quad . \tag{57}$$

The vector of curvatures $\{\chi\}$ is now rewritten as

$$\{\chi\} = \{\chi_1\} + \{\chi_2\} = ([B_2] + [B_3]) \, \{a\} \quad , \tag{58}$$

where

$$\{\chi_1\} = \begin{Bmatrix} \dfrac{\partial^2 w}{\partial x^2} \\[2ex] \dfrac{\partial^2 w}{\partial y^2} \\[2ex] 2\dfrac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} = \begin{Bmatrix} \sum a_i m_i (m_i - 1) x^{(m_i-2)} y^{n_i} \\[2ex] \sum a_i n_i (n_i - 1) x^{m_i} y^{(n_i-2)} \\[2ex] 2\sum a_i m_i n_i x^{(m_i-1)} y^{(n_i-1)} \end{Bmatrix} \tag{58a}$$

and

$$\{\chi_2\} = \begin{Bmatrix} \dfrac{-\partial \gamma_{xz}}{\partial x} \\[2ex] \dfrac{-\partial \gamma_{yz}}{\partial y} \\[2ex] \dfrac{-\partial \gamma_{xz}}{\partial y} - \dfrac{\partial \gamma_{yz}}{\partial x} \end{Bmatrix} . \tag{58b}$$

It follows that $\{\chi_1\}$ is the vector of curvature in the absence of transverse shear and $\{\chi_2\}$ is the contribution of transverse shear to the vector of curvatures.

Substituting Equations 52 and 58 into Equation 55 and comparing the resultant equation with Equation 57 and noting that {a} is independent of x and y, the generalized stiffness matrix can be obtained as

$$\begin{aligned}[K_{gen}] = &\iint [B_2]^T [D] [B_2]dxdy + \iint [B_2]^T [D] [B_3]dxdy \\ &+\iint [B_3]^T [D] [B_4]dxdy + \iint [B_3]^T [D] [B_3]dxdy \\ &+\iint [B_1]^T [G] [B_1]dxdy \quad . \end{aligned} \tag{59}$$

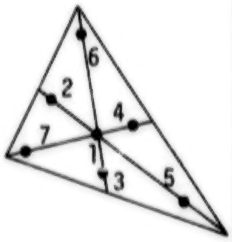
The evaluation of the elements of the generalized stiffness matrix $[k_{gen}]$ in closed form is, though straightforward, very tedious. The first term $\iint [B_2]^T [D] [B_2]dxdy$ is evaluated in closed form; the other four terms are evaluated by using numerical integration. If the transverse shear is negligible, the matrices $[B_1]$ and $[B_3]$ are null and the last four terms vanish. The numerical integration formulae used are the seven-point integration scheme (Reference 6) and are given below for easy reference. For a triangle, the integrals of the form

$$I = \int_0^1 \int_0^{1-L_1} f(L_1L_2L_3)dL_1dL_2 \tag{60}$$

can be integrated by using a seven-point numerical integration which can exactly integrate functions up to and including quintic order. The value of the integral is given by

$$I = \sum_{k=1}^{7} W_k f_k(L_1,L_2,L_3) \quad , \tag{61}$$

where the points and the weighting factors are as follows:



Point	Triangular Coordinates L_1, L_2, L_3	Weight, $2W_k$
1	1/3, 1/3, 1/3	0.225
2	α_1 β_1 β_1	0.13239415
3	β_1 α_1 β_1	
4	β_1 β_1 α_1	
5	α_2 β_2 β_2	0.12593918
6	β_2 α_2 β_2	
7	β_2 β_2 α_2	

with

$\alpha_1 = 0.05971588$ $\beta_1 = 0.47014206$

$\alpha_2 = 0.79742699$ $\beta_2 = 0.101286505$

Note the error in the value of α_1 as given in Reference 6, page 151.

Denoting by G_{11}, G_{12}, G_{13}, G_{22}, G_{23}, and G_{33} the symmetric portion of the $[G_e]$ matrix of Equation 26, it can be shown that the jth element of the ith row of the generalized stiffness matrix $[K_{gen}]$, for the case of a plate with transverse shear being neglected, is given by

$$(K_{ij})_{gen} = \frac{1}{12} \sum_{k_1=1}^{3} \sum_{k_2=1}^{3} \sum_{k_3=1}^{3} c_{k_1} c_{k_2} c_{k_3}$$

$$\Big[G_{11} m_i m_j (m_i - 1)(m_j - 1)\, F(m_i + m_j + \gamma_{k_1} + \gamma_{k_2}$$

$$+ \gamma_{k_3} - 4,\ n_i + n_j + s_{k_1} + s_{k_2} + s_{k_3})$$

$$+ G_{22} n_i n_j (n_i - 1)\ (n_j - 1)\ F(m_i + m_j + \gamma_{k_1}$$

$$
\begin{aligned}
&+ \gamma_{k_2} + \gamma_{k_3}, n_i + n_j + s_{k_1} + s_{k_2} + s_{k_3} - 4) \\
&+ (4G_{33} m_i m_j n_i n_j + G_{12}\{m_i n_j (m_i - 1)\,(n_j - 1) \\
&+ m_j n_i (m_j - 1)(n_i - 1)\}) \, F(m_i + m_j + \gamma_{k_1} \\
&+ \gamma_{k_2} + \gamma_{k_3} - 2, n_i + n_j + s_{k_1} + s_{k_2} + s_{k_3} - 2 \\
&+ 2G_{13}\{m_i m_j n_j (m_i - 1) + m_i n_i m_j (m_j - 1)\} \;\; F(m_i \\
&+ m_j + \gamma_{k_1} + \gamma_{k_2} + \gamma_{k_3} - 3, n_i + n_j + s_{k_1} \\
&+ s_{k_2} + s_{k_3} - 1 + 2G_{23}\{m_j n_i n_j (n_i - 1) \\
&+ m_i n_i n_j (n_j - 1)\} \, F(m_i + m_j + \gamma_{k_1} + \gamma_{k_2} \\
&+ \gamma_{k_3} - 1, n_i + n_j + s_{k_1} + s_{k_2} + s_{k_3} - 3) \Big] \quad . \qquad (62)
\end{aligned}
$$

All computations involved in evaluating $[K_{gen}]$ for the case of a plate with transverse shear neglected can be carried out by the close form expression (62). For plates with transverse shear flexibility, the contribution of the last four integrals of Equation 59 will be evaluated using the numerical integration formulae listed earlier and algebraically added on to the closed form expression for $[K_{gen}]$ evaluated by Equation 62.

Once the generalized stiffness matrix $[K_{gen}]$ is evaluated, the element stiffness matrix in the local element coordinates $[K_{ee}]$ is obtained, by virtue of Equation 17, as

$$[K_{ee}] = [S]^T \, [K_{gen}] \, [S] \quad . \qquad (63)$$

$[K_{ee}]$ can then be transformed to the global coordinate system in the same manner as for all other elements.

Let the transformation for displacements be

$$\{u\}_{basic} = [E]^T \{u\}_{element} \tag{64}$$

and

$$\{u\}_g = [T] \{u\}_{basic} \tag{65}$$

Then, the stiffness matrix in global coordinates is

$$[K_{gg}] = [T]^T [E] [K_{ee}] [E]^T [T] \quad . \tag{66}$$

5.8.7.6 Equivalent Thermal Loads

The stress-free strains developed in a free plate due to a variation of temperature with depth are

$$\{\epsilon_t\} = \begin{Bmatrix} \epsilon_{xt} \\ \epsilon_{yt} \\ \gamma_t \end{Bmatrix} = \begin{Bmatrix} \alpha_{e1} \\ \alpha_{e2} \\ \alpha_{e3} \end{Bmatrix} (\bar{T} - T_\gamma) = \{\alpha_e\} [\bar{T} - T_\gamma) \quad , \tag{67}$$

where $\bar{T}$ is the temperature at any point (x,y,z) of the element, T_γ is the reference or stress-free temperature of the material, and $\{\alpha_e\}$ is the vector of thermal expansion coefficients in the element coordinate system.

An applied stress vector which would produce the thermal strains is

$$\{\sigma_t\} = [G_e] \{\epsilon_t\} = [G_e] \{\alpha_e\} (\bar{T} - T_\gamma) \quad , \tag{68}$$

where $[G_e]$ is the matrix of elastic coefficients at the point on the cross-section.

The generalized equivalent thermal load vector $\{P^t_{gen}\}$ is obtained as

$$\{P^t_{gen}\} = \frac{\partial}{\partial\{a\}} \int_V \{\epsilon\}^T \{\sigma_t\} dV. \tag{69}$$

The strains $\{\epsilon\}$ are related to the curvatures $\{\chi\}$ by

$$\{\epsilon\} = -z\{\chi\} \quad , \tag{70}$$

351

where z is measured from the neutral surface of the plate. Substituting Equations 68 and 70 into Equation 69,

$$\{P^t_{gen}\} = - \frac{\partial}{\partial\{a\}} \int_V z\{\chi\}^T [G_e] \{\alpha_e\} (\bar{T} - T_\gamma) dV \tag{71}$$

The variation over the surface of the element of the mean temperature, T_o, and the thermal gradient at a cross-section, T', are both assumed as bilinear polynomials:

$$T_o = \sum_{i=1}^{3} d_i x^{P_i} y^{q_i} \tag{72}$$

and

$$T' = \sum_{i=1}^{3} d_i' x^{P_i} y^{V_i} \tag{73}$$

so that the temperature at any point (x,y,z) is

$$T = T_o + T'z \quad . \tag{74}$$

The constants d_i and d_i' are evaluated from the values at the vertices. Thus,

$$d_1 = \frac{T_{o1}'a + T_{o3}'b}{(a + b)} \tag{75}$$

$$d_2 = \frac{T_{o3}' - T_{o1}'}{(a + b)} \tag{76}$$

$$d_3 = \frac{1}{c} [T_{o5}' - d_1] \tag{77}$$

$$d_1' = \frac{T_1'a + T_3'b}{(a + b)} \tag{78}$$

$$d_2' = \frac{T_3' - T_1'}{(a + b)} \tag{79}$$

and

$$d_3' = \frac{1}{c} [T_5' - d_1'] \tag{80}$$

where T_{o1}', T_{o3}', and T_{o5}' are the differences between the grid point temperatures and the reference temperature at grid points 1, 3, and 5 respectively.

It is convenient to define the equivalent thermal moment vector

$$\{M_t\} = -\int_z [G_e]\ \{\alpha_e\}\ (\bar{T} - T_\gamma)zdz$$

$$= -\int_{-t/2}^{+t/2} [G_e]\ \{\alpha_e\}\ (T_o' + T'z)zdz$$

$$= -\ [G_e]\ \{\alpha_e\}\ T'\ \frac{t^3}{12} \quad . \tag{81}$$

Substituting for t from Equation 21a and for T' from Equation 73,

$$\{M_t\} = -\ \frac{1}{12}\ [G_e]\ \{\alpha_e\} \sum_{i_1=1}^{3}\ \sum_{i_2=1}^{3}\ \sum_{i_3=1}^{3}\ \sum_{j=1}^{3}\ c_{i_1} c_{i_2} c_{i_3} d_j'$$

$$x^{(\gamma_{i_1}+\gamma_{i_2}+\gamma_{i_3}+P_j)}\ y^{(s_{i_1}+s_{i_2}+s_{i_3}+V_j)} \quad . \tag{82}$$

At the three vertices, the value of $\{M_t\}$ will be given by

$$\{M_t\}_1 = -[G_e]\ \{\alpha_e\}\ I_1 T_1' \tag{83}$$

$$\{M_t\}_3 = -[G_e]\ \{\alpha_e\}\ I_3 T_3' \tag{84}$$

and

$$\{M_t\}_5 = -[G_e]\ \{\alpha_e\}\ I_5 T_5' \quad , \tag{85}$$

where I_1, I_3, and I_5 are the moments of inertia of the cross section at the vertices G_1, G_3, and G_5 of the element, respectively. The "effective" thermal gradient, T', at the vertices is given by

$$T_1' = \frac{1}{I_1}\int T_1 zdz \tag{86}$$

$$T_3' = \frac{1}{I_3}\int T_3 zdz \tag{87}$$

and

$$T_5' = \frac{1}{I_5}\int T_5 zdz \quad , \tag{88}$$

and the integration is carried out over the effective bending material. The user must specify the thermal gradient at the three vertices, if the temperature is to have bilinear variation over the surface of the element. For solid homogeneous plates the option is provided to specify the temperature at the vertices as a tabular function of depth, in which case the thermal gradients at the three vertices will be evaluated by the program using Equations 86, 87, and 88.

Substituting Equations 21a, 58, 72, 73, and 74 into Equation 71,

$$\{P^t_{gen}\} = -\frac{1}{12}\frac{\partial}{\partial\{a\}}\iint (\{\chi_1\} + \{\chi_2\})^T [G_e] \{\alpha_e\}$$

$$\sum_{i_1=1}^{3}\sum_{i_2=1}^{3}\sum_{i_3=1}^{3}\sum_{j=1}^{3} c_{i_1} c_{i_2} c_{i_3} d'_j$$

$$x^{(\gamma_{i_1}+\gamma_{i_2}+\gamma_{i_3}+P_j)}\ y^{(s_{i_1}+s_{i_2}+s_{i_3}+V_j)}\ dxdy \quad . \tag{89}$$

As in the case of the derivation of generalized stiffness matrix, the generalized thermal load vector will be evaluated in two stages, viz., the closed form expression $[P^t_{gen}]_1$ due to $[\chi_1]$, the vector of curvatures in the absence of transverse shear, and the numerically integrated expression $[P^t_{gen}]_2$ due to $[\chi_2]$, the contribution of transverse shear to the vector of curvatures. Using the following notations, viz.,

$$G'_{11} = G_{11}\alpha_{e_1} + G_{12}\alpha_{e_2} + G_{13}\alpha_{e_3} \tag{90}$$

$$G'_{22} = G_{12}\alpha_{e_1} + G_{22}\alpha_{e_2} + G_{23}\alpha_{e_3} \tag{91}$$

and

$$G'_{33} = G_{13}\alpha_{e_1} + G_{23}\alpha_{e_2} + G_{33}\alpha_{e_3} \quad , \tag{92}$$

the ith element of the generalized load vector $\{P^t_{gen}\}$ will be given by

$$
\begin{aligned}
\{P^t_{gen}\}_1 = \frac{1}{12} \sum_{i_1=1}^{3} \sum_{i_2=1}^{3} \sum_{i_3=1}^{3} \sum_{j=1}^{3} c_{i_1} c_{i_2} c_{i_3} d'_j \\
[G'_{11} m_i (m_i - 1)\ F(m_i + \gamma_{i_1} + \gamma_{i_2} + \gamma_{i_3} + P_j - 2, \\
n_i + s_{i_1} + s_{i_2} + s_{i_3} + V_j) + G'_{22}\ n_i (n_i - 1) \\
F(m_i + \gamma_{i_1} + \gamma_{i_2} + \gamma_{i_3} + P_j,\ n_i + s_{i_1} + s_{i_2} \\
+ s_{i_3} + V_j - 2) + G'_{33} m_i n_i\ F(m_i + \gamma_{i_1} + \gamma_{i_2} \\
+ \gamma_{i_3} + P_j - 1,\ n_i + s_{i_1} + s_{i_2} + s_{i_3} + V_j - 1)] \quad . \qquad (93)
\end{aligned}
$$

The load vector $\{P^t_{gen}\}_2$ is evaluated using numerical integration and $[P^t_{gen}]$ is obtained as the sum of $[P^t_{gen}]_1$ and $[P^t_{gen}]_2$. For plates with negligible transverse shear, $[P^t_{gen}]_2$ is null. The equivalent thermal bending load in the local element coordinate system is obtained, by virtue of Equation 17, as

$$\{P^t_e\} = [S]^T\ \{P^t_{gen}\} \quad . \qquad (94)$$

The load vector can then be transformed to the global system by

$$\{P^t_g\} = [T]^T\ [E]\ \{P^t_e\} \quad . \qquad (95)$$

5.8.7.7 Recovery of Internal Forces

The internal forces are recovered at the three vertices and at the centroid of the element. After the displacements of the element are transformed from the global system $\{u\}_g$ to the element coordinate system $\{u\}_e$, the generalized coordinates $\{a\}$ are evaluated from Equation 17. The curvatures $\{\chi\}$ are evaluated from Equation 58 with the nonzero elements of $[B_3]$ being as listed below:

$$B_3(1,11) = -24A_{11}$$

$$B_3(1,12) = -6A_{31}$$

$B_3(1,13) = -4A_{32}$

$B_3(1,14) = -6A_{15}$

$B_3(1,16) = -120A_{11}x$

$B_3(1,17) = -24(A_{31}x + A_{11}y)$

$B_3(1,18) = -12(A_{32}x + A_{31}y)$

$B_3(1,19) = -12(A_{15}x + A_{32}y)$

$B_3(1,20) = -24A_{15}y$

$B_3(2,12) = -6A_{21}$

$B_3(2,13) = -4A_{33}$

$B_3(2,14) = -6A_{34}$

$B_3(2,15) = -24A_{25}$

$B_3(2,17) = -24A_{21}x$

$B_3(2,18) = -12(A_{33}x + A_{21}y)$

$B_3(2,19) = -12(A_{34}x + A_{33}y)$

$B_3(2,20) = -24(A_{25}x + A_{34}y)$

$B_3(2,21) = -120A_{25}y$

$B_3(3,11) = -24A_{21}$

$B_3(3,12) = -6(A_{11} + A_{33})$

$B_3(3,13) = -4(A_{31} + A_{34})$

$B_3(3,14) = -6(A_{32} + A_{25})$

$B_3(3,15) = -24A_{15}$

$B_3(3,16) = -120A_{21}x$

$B_3(3,17) = -24[(A_{11} + A_{33})x + A_{21}y]$

$B_3(3,18) = -12[(A_{34} + A_{31})x + (A_{34} + A_{11})y]$

$B_3(3,19) = -12[(A_{25} + A_{32})x + (A_{34} + A_{31})y]$

$$B_3(3,20) = -24[A_{15}x + (A_{32} + A_{25})y]$$

and

$$B_3(3,21) = -120A_{15}y \quad ,$$

where $A_{11}, A_{12}, \ldots, A_{34}$ are as given in Equations 38 and 53. Moments at the vertices are then obtained from

$$\{M\}_1 = [D]_1 \{\chi\} - \{M_t\}_1 \tag{96}$$

$$\{M\}_3 = [D]_3 \{\chi\} - \{M_t\}_3 \tag{97}$$

and

$$\{M\}_5 = [D]_5 \{\chi\} - \{M_t\}_5 \quad . \tag{98}$$

The moment at the centroid is evaluated from similar expressions as Equations 96, 97, and 98, with the value of [D] and $\{M_t\}$ evaluated from Equations 25 and 82 respectively.

The transverse shears are evaluated as follows:

$\{\gamma\}$ is evaluated from Equations 52, 53, and 17.

{V} is then evaluated from Equations 29 and 30.

REFERENCES

1. Narayanaswami, R., "New Triangular and Quadrilateral Plate-Bending Finite Elements," NASA TN D-7407, April 1974.

2. Narayanaswami, R., "New Triangular Plate-Bending Finite Element with Transverse Shear Flexibility," AIAA Journal, December 1974, pp. 1761-1763.

3. Cowper, G. R., Kosko, E., Lindberg, G. M., and Olson, M. D., "A High-Precision Triangular Plate-Bending Element," Aeronautical Report LR-514, National Research Council, Ottawa, Canada, December 1968.

4. Bell, K., "Triangular Plate Bending Elements," in Finite Element Methods in Stress Analysis, Ivar Holland and Kolbein Bell, eds., TAPIR (Trondheim, Norway), 1969, pp. 213-252.

5. Timoshenko, S., and Woinowsky-Krieger, S., Theory of Plates and Shells, Second Ed., McGraw-Hill Book Co., Inc., 1959.

6. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill Book Co., 1971.

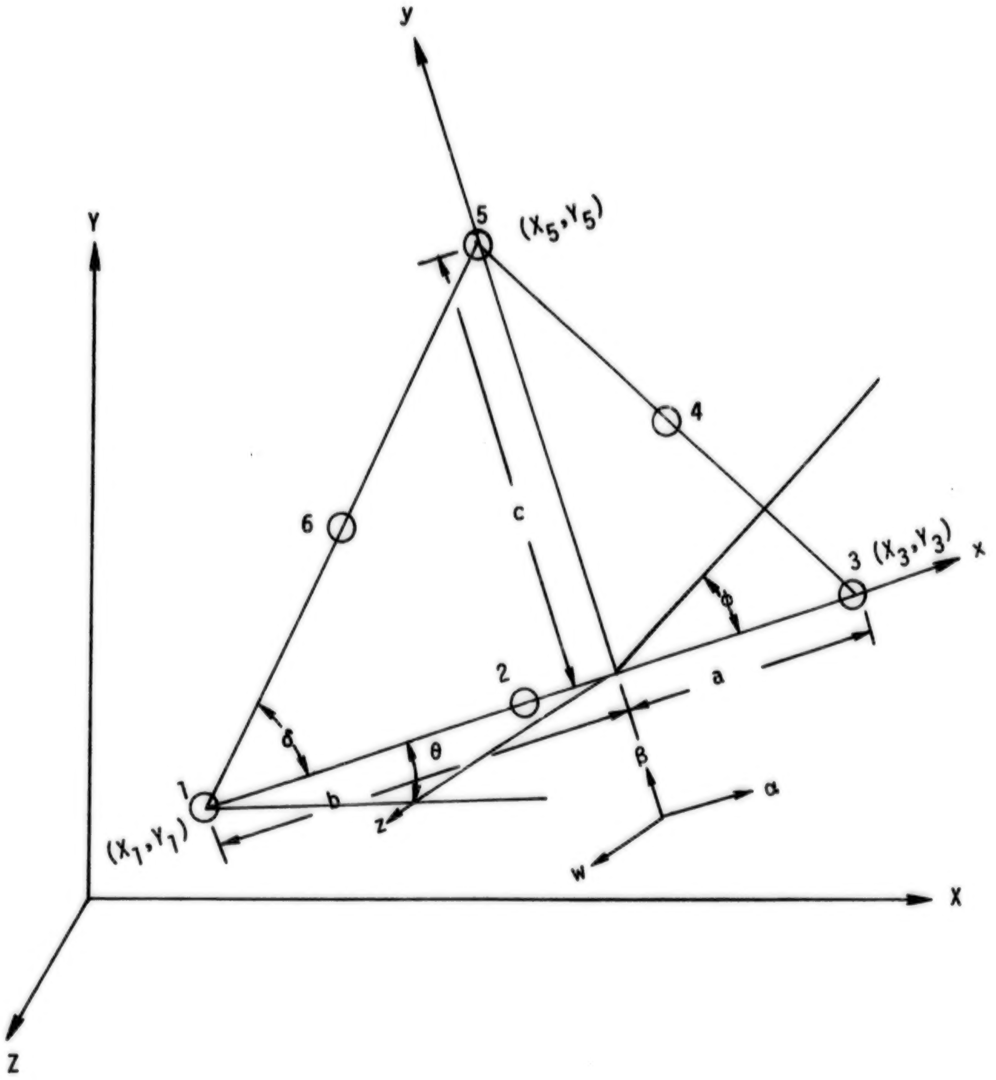

y
Y
5
(X_5, Y_5)
4
6
c
3 (X_3, Y_3)
x
2
a
ϕ
δ
θ
1
(X_1, Y_1)
b
z
β
α
w
X
Z

Figure 1. TRPLT1 triangular bending element geometry.

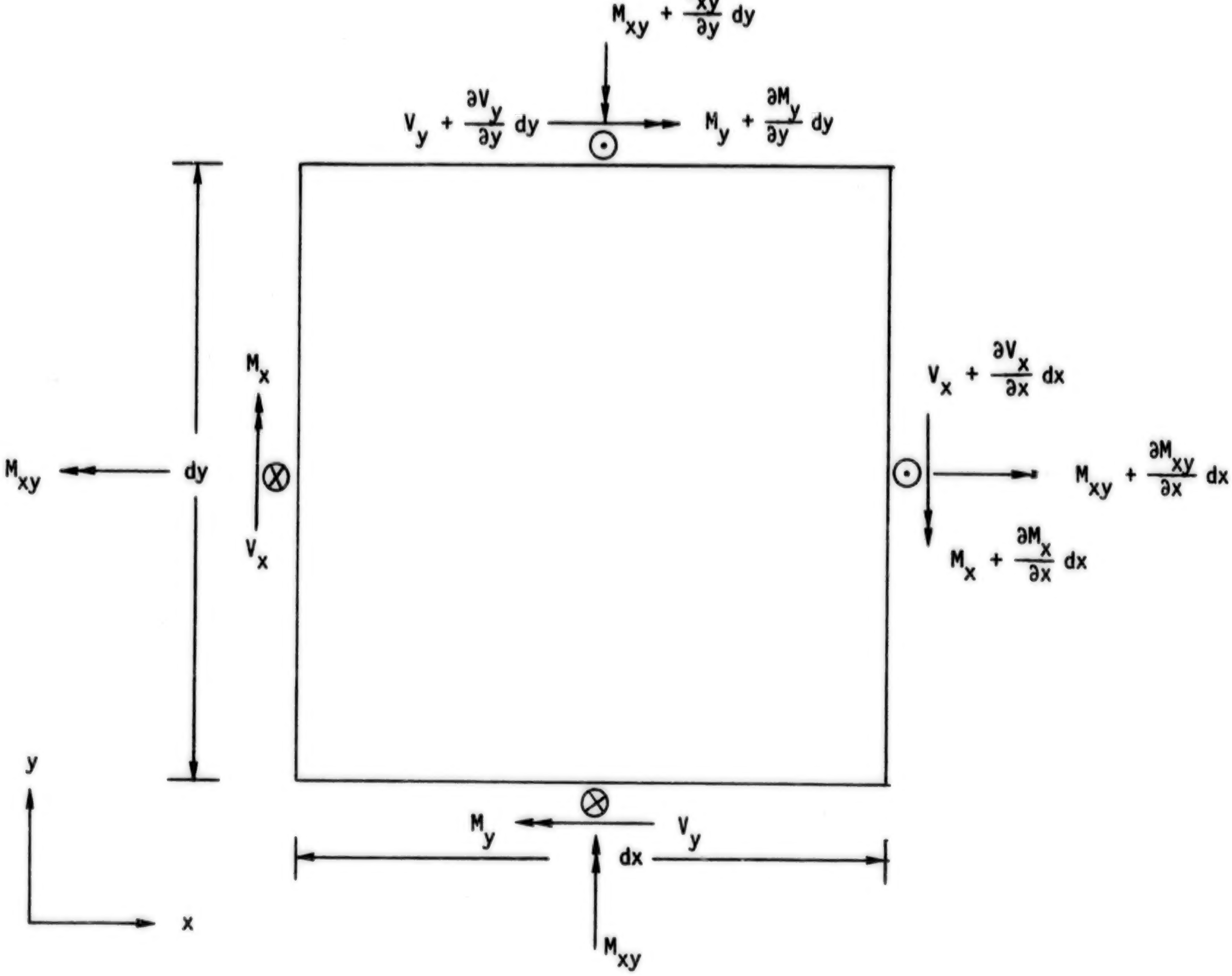

$M_{xy} + \frac{\partial_{xy}}{\partial y} dy$
$V_y + \frac{\partial V_y}{\partial y} dy$
$M_y + \frac{\partial M_y}{\partial y} dy$
M_x
M_{xy}
dy
V_x
$V_x + \frac{\partial V_x}{\partial x} dx$
$M_{xy} + \frac{\partial M_{xy}}{\partial x} dx$
$M_x + \frac{\partial M_x}{\partial x} dx$
y
x
M_y
V_y
dx
M_{xy}

Figure 2. Sign convention for moments and shears.

5.9 THE CONICAL SHELL ELEMENT

The properties of the conical shell element are assumed to be symmetrical with respect to the axis of the shell. The loads and deflections, on the other hand, need not be axisymmetric; they are expanded in Fourier series with respect to the azimuth coordinate. Due to symmetry, the resulting load and deformation systems for different harmonic orders are independent, a fact that results in large time saving when the use of the conical shell element is compared with an equivalent model constructed from plate elements.

Equations for the element are developed in terms of Fourier coefficients with respect to azimuth and in terms of polynomial coefficients with respect to meridional distance. An important and unusual feature of the NASTRAN conical shell element is that it includes transverse shear flexibility. At present the conical shell element cannot be combined with other types of structural elements in the solution of problems.

5.9.1 Coordinate Notation

The coordinate geometry for the conical shell element is shown in Figure 1. The internal coordinate system for the element is oriented in and normal to the surface of the shell. The coordinate system for grid points at the ends of the element will usually be parallel and perpendicular to the axis of the shell.

Stiffness matrices will be derived in terms of element coordinates evaluated at the ends of the element. The stiffness matrices must then be transformed into the global coordinate system, which matter is not treated here.

Although the general case of a conical shell is treated, the results obtained are valid for the limiting cases of a cylinder, $\psi = 0$, and of a flat circular plate, $\psi = \pi/2$.

5.9.2 Harmonic Dependence on Azimuth Position

Since the conical shell element is assumed to be axisymmetric, the motions of the shell at meridional position, s, can be expanded in a trigonometric series with respect to azimuth position, ϕ:

PRECEDING PAGE BLANK NOT FILMED

PAGE 301 INTENTIONALLY BLANK

$$\left.\begin{aligned}
u(s,\phi) &= \sum_{n=1}^{m} u_n(s)\sin(n\phi) + u_o^*(s) - \sum_{n=1}^{m} u_n^*(s)\cos(n\phi) \quad , \\
v(s,\phi) &= v_o(s) + \sum_{n=1}^{m} v_n(s)\cos(n\phi) + \sum_{n=1}^{m} v_n^*(s)\sin(n\phi) \quad , \\
w(s,\phi) &= w_o(s) + \sum_{n=1}^{m} w_n(s)\cos(n\phi) + \sum_{n=1}^{m} w_n^*(s)\sin(n\phi) \quad , \\
\alpha(s,\phi) &= \alpha_o(s) + \sum_{n=1}^{m} \alpha_n(s)\cos(n\phi) + \sum_{n=1}^{m} \alpha_n^*(s)\sin(n\phi) \quad , \\
\beta(s,\phi) &= \sum_{n=1}^{m} \beta_n(s)\sin(n\phi) + \beta_o^*(s) - \sum_{n=1}^{m} \beta_n^*(s)\cos(n\phi) \quad .
\end{aligned}\right\} \qquad (1)$$

The rotations α and β are independent motions because of the transverse shear flexibility. Rotation about the normal to the surface is not included, such rotation being adequately represented by the gradients of u and v.

5.9.3 Cases to be Treated

The motions corresponding to different harmonic orders (different values of n) are elastically uncoupled. Furthermore, motions represented by starred parameters are not coupled to motions represented by unstarred parameters. For $n > 0$ the stiffness matrices for the starred and unstarred motions are identical. The reason is that the starred parameters describe motions that are all shifted $\frac{\pi}{2n}$ in azimuth from the motions described by the unstarred parameters. Thus, $-\cos(n\phi) = \sin(n\phi - \frac{\pi}{2})$ and $\sin(n\phi) = \cos(n\phi - \frac{\pi}{2})$. The unstarred motions will be used to develop the stiffness matrices for $n > 0$.

The set of parameters, $v_o(s)$, $w_o(s)$ and $\alpha_o(s)$ describes axisymmetric motion of the shell. The set of parameters, $u_o^*(s)$ and $\beta_o^*(s)$, describes rotation and twisting of the shell about its axis. The stiffness matrix for $n = 0$ will include both starred and unstarred motions.

The degrees of freedom for the shell element are taken to be the values of the Fourier coefficients appearing in Equation 1, evaluated at the ends of the shell element. Separate stiffness matrices will be evaluated for the following parameter sets,

$$\{u_{eo}\}^T = \lfloor u_{ao}^*, v_{ao}, w_{ao}, \alpha_{ao}, \beta_{ao}^*; u_{bo}^*, v_{bo}, w_{bo}, \alpha_{bo}, \beta_{bo}^* \rfloor \quad , \qquad (2)$$

$$\{u_{e1}\}^T = \lfloor u_{a1}, v_{a1}, w_{a1}, \alpha_{a1}, \beta_{a1}; u_{b1}, v_{b1}, w_{b1}, \alpha_{b1}, \beta_{b1} \rfloor \ . \qquad (3)$$

and in general

$$\{u_{en}\}^T = \lfloor u_{an}, v_{an}, w_{an}, \alpha_{an}, \beta_{an}; u_{bn}, v_{bn}, w_{bn}, \alpha_{bn}, \beta_{bn} \rfloor \ , \qquad (4)$$

where subscripts a and b refer to the ends of the segment, see Figure 1. Note that, as previously shown, the stiffness matrices for the starred components are identical to those for the unstarred components, for $n > 0$, and need not be separately calculated. The general starred parameter set is

$$\{u^*_{en}\}^T = \lfloor u^*_{an}, v^*_{an}, w^*_{an}, \alpha^*_{an}, \beta^*_{an}; u^*_{bn}, v^*_{bn}, w^*_{bn}, \alpha^*_{bn}\ \beta^*_{bn} \rfloor \ . \qquad (5)$$

If transverse shear flexibility is negligible, the rotations, β_a and β_b, are not independent degrees of freedom. Special procedures are required for the case of zero transverse shear flexibility. Stiffness matrices will be separately derived for the following cases, in the following order. Note that the stiffness matrices for $n > 0$ can be derived with n as a parameter.

a. Finite shear flexibility, $n > 0$.

b. Finite shear flexibility, $n = 0$.

c. Zero shear flexibility, $n > 0$.

d. Zero shear flexibility, $n = 0$.

5.9.4 General Plan for Deriving the Stiffness Matrices

For each harmonic index the displacements of the shell are approximated by power series with respect to distance along the shell. The power series include a number of independent constants equal to the number of degrees of freedom. For example, the general case of finite shear flexibility and $n > 0$ requires ten independent constants, i.e. one for each element of $\{u_{en}\}$. The relationship between degrees of freedom and the independent constants, $\{q_n\}$, can be explicitly stated as

$$\{u_{en}\} = [H_{uq}]\{q_n\} \ . \qquad (6)$$

The next step is to express strains in terms of the independent constants. The strains, of course, have harmonic dependence on azimuth similar to Equation 1, so that the required relationships are between harmonic coefficients of strain and the independent constants for the same harmonic. Three kinds are involved.

$$\{\varepsilon\} = [\varepsilon_s, \varepsilon_\phi, \varepsilon_{s\phi}]^T, \text{ membrane strains} \quad , \tag{7}$$

$$\{\gamma\} = [\gamma_s, \gamma_\phi]^T, \text{ transverse shear strains, and} \tag{8}$$

$$\{\chi\} = [\chi_s, \chi_\phi, \chi_{s\phi}]^T, \text{ bending curvatures.} \tag{9}$$

The required relationships are:

$$\{\varepsilon_n\} = [H_{\varepsilon q}]_n\{q_n\} \quad , \tag{10}$$

$$\{\gamma_n\} = [H_{\gamma q}]_n\{q_n\} \quad , \tag{11}$$

$$\{\chi_n\} = [H_{\chi q}]_n\{q_n\} \quad . \tag{12}$$

The matrix coefficients are evaluated by combining the relationship between strains and displacements with the relationship between displacements and the independent constants.

The total strain energy for the conical shell element is

$$V_e = \frac{1}{2}\int_0^\ell \int_0^{2\pi} [\{F\}^T\{\varepsilon\} + \{V\}^T\{\gamma\} + \{M\}^T\{\chi\}]\ r\,d\phi\,ds \quad , \tag{13}$$

where, for a unit width of shell,

$$\{F\}^T = [F_s, F_\phi, F_{s\phi}], \text{ membrane forces} \quad , \tag{14}$$

$$\{V\}^T = [V_s, V_\phi], \text{ transverse shear forces, and} \tag{15}$$

$$\{M\}^T = [M_s, M_\phi, M_{s\phi}], \text{ bending and twisting moments} \ . \tag{16}$$

Forces and moments are related to strains by elasticity. It is assumed that the three types of strains are uncoupled so that

$$\{F\} = t[E]\{\varepsilon\} \quad , \tag{17}$$

$$\{V\} = t_s[G]\{\gamma\} \quad , \tag{18}$$

$$\{M\} = [D]\{\chi\} \quad , \tag{19}$$

where t is the thickness of the shell for membrane stiffness, and t_s is the thickness of the shell for transverse shear stiffness, both assumed constant over the surface of the element. Because of the symmetry of the shell, certain terms in [E], [G] and [D] are zero. The remaining terms, assumed to be constant over the surface of the element, are

$$[E] = \begin{bmatrix} E_{11} & E_{12} & 0 \\ E_{12} & E_{22} & 0 \\ 0 & 0 & E_{33} \end{bmatrix} , \qquad (20)$$

$$[G] = \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix} , \qquad (21)$$

and

$$[D] = \begin{bmatrix} D_{11} & D_{12} & 0 \\ D_{12} & D_{22} & 0 \\ 0 & 0 & D_{33} \end{bmatrix} . \qquad (22)$$

By substituting from Equations 17, 18 and 19 into Equation 13, we obtain:

$$V_e = \frac{1}{2}\int_0^{\ell}\int_0^{2\pi}[t\{\varepsilon\}^T[E]\{\varepsilon\} + t_s\{\gamma\}^T[G]\{\gamma\} + \{\chi\}^T[D]\{\chi\}]\ rd\phi ds \ . \qquad (23)$$

Because of the assumed symmetry, the strain energy can be written as the sum of a series of independent terms

$$V_e = V_{eo} + \sum_{n=1}^{m} V_{en} + V_{eo}^{*} + \sum_{n=1}^{m} V_{en}^{*} , \qquad (24)$$

where each term has the form given by Equation 23, with the addition of a $\sin^2 n\phi$ or a $\cos^2 n\phi$ factor, and the substitution of harmonic strain coefficients for the strains. Integration with respect to ϕ then gives, for $n = 0$

$$V_{eo} = \pi\int_0^{\ell}[t\{\varepsilon_o\}^T[E]\{\varepsilon_o\} + t_s\{\gamma_o\}^T[G]\{\gamma_o\} + \{\chi_o\}^T[D]\{\chi_o\}]\ rds \ , \qquad (25)$$

and, for n > 0

$$V_{en} = \frac{\pi}{2}\int_0^{\ell}[t\{\varepsilon_n\}^T[E]\{\varepsilon_n\} + t_s\{\gamma_n\}^T[G]\{\gamma_n\} + \{\chi_n\}^T[D]\{\chi_n\}]\ rds \quad , \qquad (26)$$

and similar results for the starred terms.

Substitution of Equations 10, 11 and 12 into Equation 26 then gives

$$V_{en} = \frac{1}{2}\{q_n\}^T[K_n^q]\{q_n\} \quad , \qquad (27)$$

where

$$[K_n^q] = \pi\int_0^{\ell}[t[H_{\varepsilon q}]_n{}^T[E][H_{\varepsilon q}]_n + t_s[H_{\gamma q}]_n{}^T[G][H_{\gamma q}]_n + [H_{\chi q}]_n{}^T[D][H_{\chi q}]_n]\ rds. \qquad (28)$$

The result for n = 0 is the same except that the factor π is replaced by 2π.

$[K_n^q]$ is the stiffness matrix referred to the independent constants. The stiffness matrix referred to the degrees of freedom for the element is

$$[K_n^u] = [H_{uq}{}^{-1}]_n^T[K_n^q][H_{uq}]_n^{-1} \quad , \qquad (29)$$

which is the final result.

The plan of the analysis is to develop explicit formulas for the terms in $[H_{uq}]_n$ and $[K_n^q]$. The integration indicated in Equation 28 makes it very difficult, if not impossible, to express $[K_n^q]$ as a sum of products of elementary matrices.

5.9.5 Stiffness Matrix for Finite Shear Flexibility, n > 0

5.9.5.1 Power Series Expansions

The ten independent constants for each harmonic order, $\{q_n\}$, are chosen to be the coefficients in the power series expansions for displacement and shear strain amplitude coefficients as shown below. Shear strains rather than rotations are employed for convenience.

$$\left.\begin{array}{l} u_n(s) = q_{1n} + q_{2n}s \quad , \\ v_n(s) = q_{3n} + q_{4n}s \quad , \\ w_n(s) = q_{5n} + q_{6n}s + q_{7n}s^2 + q_{8n}s^3 \ . \end{array}\right\} \begin{array}{l}\text{Harmonic Components}\\ \text{of Deflection}\end{array} \qquad \begin{array}{r}(30)\\(31)\\(32)\end{array}$$

$$\gamma_{\phi n}(s) = q_{9n} + q_{10n}s \quad , \tag{33}$$

γ_{sn} is assumed constant and independent of s. The magnitude of γ_{sn} is determined in terms of the ten independent constants.

} Harmonic Components of Transverse Shear Strains

Other choices of expansions are possible. The above has been selected by virtue of the following arguments.

a. Only four constants should be associated with u_n and v_n because, in the limiting case of of a flat circular plate, u and v become uncoupled from the other degrees of freedom.

b. The choice of a cubic expansion for w is analogous to the expansions used for the transverse deflection of beam and plate elements.

c. The identity of two of the independent constants with the circumferential shear strains is necessary because, in the limiting case of zero circumferential shear flexibility, β_a and β_b are no longer independent degrees of freedom. In this limit q_{9n} and q_{10n} are omitted.

5.9.5.2 Strain-Displacement Relationships

The strain-displacement relationships for a conical shell are as follows (refer to Figure 1):

a. Membrane Strains

$$\varepsilon_s = \frac{\partial v}{\partial s} \quad , \tag{34}$$

$$\varepsilon_\phi = \frac{1}{r}\left(\frac{\partial u}{\partial \phi} + v\sin\psi + w\cos\psi\right) \quad , \tag{35}$$

$$\varepsilon_{s\phi} = \frac{\partial u}{\partial s} - \frac{1}{r}\left(u\sin\psi - \frac{\partial v}{\partial \phi}\right) \quad . \tag{36}$$

b. Transverse Shear Strains

$$\gamma_s = \frac{\partial w}{\partial s} - \alpha \quad , \tag{37}$$

$$\gamma_\phi = \frac{1}{r}\frac{\partial w}{\partial \phi} - \frac{u}{r}\cos\psi + \beta \quad . \tag{38}$$

368

c. Bending Curvatures

$$\chi_s = \frac{\partial\alpha}{\partial s} \quad , \qquad (39)$$

$$\chi_\phi = \frac{1}{r}\left(-\frac{\partial\beta}{\partial\phi} + \alpha\sin\psi\right) \quad , \qquad (40)$$

$$\chi_{s\phi} = -\frac{\partial\beta}{\partial s} + \frac{1}{r}\left(\frac{\partial\alpha}{\partial\phi} + \beta\sin\psi + \theta_w\cos\psi\right) \quad , \qquad (41)$$

where θ_w is the rotation about a normal to the shell surface,

$$\theta_w = -\frac{1}{2}\left(\frac{\partial u}{\partial s} + \frac{u\sin\psi}{r} - \frac{1}{r}\frac{\partial v}{\partial\phi}\right) \quad . \qquad (42)$$

Note that the conventions for ε_s and ε_ϕ give positive strains for tension and that the conventions selected for curvatures are such that the fiber strains due to bending and stretching are algebraically added on the inner surface of the shell.

The above strain-displacement relationships are, with the exception of the twisting strain, fairly standard and they can be checked by reference to text books or to simple diagrams. For the derivation of the twisting strain refer to Figure 2. The twisting strain is defined by

$$\chi_{s\phi} = \frac{\partial\bar{\alpha}}{r\partial\phi} - \frac{\partial\beta}{\partial s} \quad . \qquad (43)$$

$\bar{\alpha}$ rather than α is used because $\frac{\partial\alpha}{\partial\phi}$ is not zero in a rigid body rotation of a surface element about the normal to the surface. $\bar{\alpha}$ is defined to be collinear on opposite edges of the element. From Figure 2b:

$$\bar{\alpha} = \alpha\cos\Delta\phi + (\theta_w\cos\psi + \beta\sin\psi)\sin\Delta\phi \quad , \qquad (44)$$

so that

$$\frac{\partial\bar{\alpha}}{\partial\phi} = \frac{\partial\alpha}{\partial\phi} + (\theta_w\cos\psi + \beta\sin\psi) \quad . \qquad (45)$$

Substitution of Equation 45 into Equation 43 gives Equation 41. Equation 42 for the rotation about the normal can be readily verified by simple diagrams.

For the unstarred parameter sets, the components of strain have the following dependence on azimuth, assuming only one harmonic order, the n^{th}, to be present. The choice of a sine or cosine dependence is made from a consideration of symmetry.

$$\begin{Bmatrix} \epsilon_s \\ \epsilon_\phi \\ \gamma_s \\ \chi_s \\ \chi_\phi \end{Bmatrix} = \cos(n\phi) \begin{Bmatrix} \epsilon_{sn} \\ \epsilon_{\phi n} \\ \gamma_{sn} \\ \chi_{sn} \\ \chi_{\phi n} \end{Bmatrix} \quad ; \quad \begin{Bmatrix} \epsilon_{s\phi} \\ \gamma_\phi \\ \chi_{s\phi} \end{Bmatrix} = \sin(n\phi) \begin{Bmatrix} \epsilon_{s\phi n} \\ \gamma_{\phi n} \\ \chi_{s\phi n} \end{Bmatrix} \quad . \qquad (46)$$

The relationships between the Fourier components of strain and the Fourier components of displacement are derived by inserting Equation (46) into the strain-displacement relations. The subscript (n), which modifies nearly every dependent variable from now in section 5.9.5, is dropped for convenience.

a. Membrane Strain Components:

$$\varepsilon_s = \frac{\partial v}{\partial s} \quad , \qquad (47)$$

$$\varepsilon_\phi = \frac{1}{r}\left(nu + v\sin\psi + w\cos\psi\right) \quad , \qquad (48)$$

$$\varepsilon_{s\phi} = \frac{\partial u}{\partial s} - \frac{1}{r}\left(u\sin\psi + nv\right) \quad . \qquad (49)$$

b. Transverse Shear Strain Components:

$$\gamma_s = \frac{\partial w}{\partial s} - \alpha \quad , \qquad (50)$$

$$\gamma_\phi = -\frac{1}{r}\left(nw + u\cos\psi\right) + \beta \quad . \qquad (51)$$

c. Bending Curvature Components:

$$\chi_s = \frac{\partial \alpha}{\partial s} \quad , \qquad (52)$$

$$\chi_\phi = \frac{1}{r}\left(-n\beta + \alpha\sin\psi\right) \quad , \qquad (53)$$

$$\chi_{s\phi} = -\frac{\partial \beta}{\partial s} + \frac{1}{r}\left[-n\alpha + \beta\sin\psi - \frac{1}{2}\cos\psi\left(\frac{\partial u}{\partial s} + \frac{u\sin\psi}{r} + \frac{n}{r}v\right)\right] \quad . \qquad (54)$$

5.9.5.3 Development of $[H_{uq}]$

Equations 50 and 51 are used to obtain the relationships between α,β and the independent constants. Substituting from Equations 30 through 33 into Equations 50 and 51, and dropping subscripts (n),

$$\alpha = \frac{\partial w}{\partial s} - \gamma_s = q_6 + 2q_7 s + 3q_8 s^2 - \gamma_s \quad , \qquad (55)$$

$$\beta = \frac{1}{r}(nw + u\cos\psi) + \gamma_\phi$$

$$= \frac{1}{r}[n(q_5 + q_6 s + q_7 s^2 + q_8 s^3) + \cos\psi(q_1 + q_2 s)] + q_9 + q_{10}s \quad . \qquad (56)$$

In order to express α in terms of the independent constants, γ_s must first be expressed in terms of the independent constants. Writing Equations 30 through 32 and 55, 56 in matrix notation yields

$$\{u_e\} = [\overline{H}_{uq}]\{q\} + \{H_{u\gamma}\}\gamma_s \quad , \qquad (57)$$

where the bar notation is used to indicate a subset of the complete $[H_{uq}]$ matrix. γ_s will be found in terms of $\{q\}$ later. From Equations 8 and 11

$$\gamma_s = \lfloor H_{\gamma_s q} \rfloor \{q\} \quad , \qquad (58)$$

where $\lfloor H_{\gamma_s q} \rfloor$ is a partition of $[H_{\gamma q}]$ in Equation 11. Thus, comparing with Equation 6

$$[H_{uq}] = [\overline{H}_{uq}] + \{H_{u\gamma}\}\lfloor H_{\gamma_s q} \rfloor \quad . \qquad (59)$$

The nonzero terms in the $[\overline{H}_{uq}]$ matrix are, recalling the order of displacement components in Equation 3:

u_a: $\overline{H}_{11} = 1$.

v_a: $\overline{H}_{23} = 1$.

w_a: $\overline{H}_{35} = 1$.

α_a: $\overline{H}_{46} = 1$.

β_a: $\overline{H}_{51} = \frac{\cos\psi}{r_a}$; $\overline{H}_{55} = \frac{n}{r_a}$; $\overline{H}_{59} = 1$.

u_b: $\overline{H}_{61} = 1$; $\overline{H}_{62} = \ell$.

v_b: $\overline{H}_{73} = 1$; $\overline{H}_{74} = \ell$.

w_b: $\overline{H}_{85} = 1$; $\overline{H}_{86} = \ell$; $\overline{H}_{87} = \ell^2$; $\overline{H}_{88} = \ell^3$.

α_b: $\overline{H}_{96} = 1$; $\overline{H}_{97} = 2\ell$; $\overline{H}_{98} = 3\ell^2$.

371

$$\beta_b:\ \bar{H}_{10,1} = \frac{\cos\psi}{r_b};\ \bar{H}_{10,2} = \frac{\ell\cos\psi}{r_b};\ \bar{H}_{10,5} = \frac{n}{r_b};\ \bar{H}_{10,6} = \frac{n\ell}{r_b};$$

$$\bar{H}_{10,7} = \frac{n\ell^2}{r_b};\ \bar{H}_{10,8} = \frac{n\ell^3}{r_b};\ \bar{H}_{10,9} = 1;\ \bar{H}_{10,10} = \ell \quad .$$

$\{H_{u\gamma}\}$ is a column vector with -1 in the 4th and 9th elements and zero elsewhere.

5.9.5.4 Development of $[H_{\varepsilon q}]$ and $[H_{\chi q}]$

From Equations 47 and 31:

$$\varepsilon_s = q_4 \quad . \tag{60}$$

From Equations 48 and 30 to 32:

$$\varepsilon_\phi = \frac{1}{r}\left[n(q_1 + q_2 s) + \sin\psi(q_3 + q_4 s) + \cos\psi(q_5 + q_6 s + q_7 s^2 + q_8 s^3)\right] \quad . \tag{61}$$

From Equations 49, 30 and 31:

$$\varepsilon_{s\phi} = q_2 - \frac{1}{r}\left[\sin\psi(q_1 + q_2 s) + n(q_3 + q_4 s)\right] \quad . \tag{62}$$

Hence:

$$[H_{\varepsilon q}]^T = \begin{array}{cccc} \varepsilon_s & \varepsilon_\phi & \varepsilon_{s\phi} & \\ \left[\begin{array}{ccc} 0 & \frac{n}{r} & \frac{-\sin\psi}{r} \\ 0 & \frac{ns}{r} & 1 - \frac{s}{r}\sin\psi \\ 0 & \frac{\sin\psi}{r} & \frac{-n}{r} \\ 1 & \frac{s\sin\psi}{r} & \frac{-ns}{r} \\ 0 & \frac{\cos\psi}{r} & 0 \\ 0 & \frac{s\cos\psi}{r} & 0 \\ 0 & \frac{s^2\cos\psi}{r} & 0 \\ 0 & \frac{s^3\cos\psi}{r} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right. & & & \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right] \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \end{array} \tag{63}$$

For the bending strains we consider the dependence on the independent constants and on γ_s separately. Thus

$$\{\chi\} = [\overline{H}_{\chi q}]\{q\} + \{H_{\chi\gamma}\}\gamma_s = [H_{\chi q}]\{q\} \qquad , \qquad (64)$$

so that using Equation 58

$$[H_{\chi q}] = [\overline{H}_{\chi q}] + \{H_{\chi\gamma}\}\lfloor H_{\gamma_s q}\rfloor \qquad . \qquad (65)$$

From Equations 52 and 55 and the assumption that γ_s is constant,

$$\chi_s = 2q_7 + 6q_8 s \qquad . \qquad (66)$$

From Equations 53, 55 and 56

$$\begin{aligned}
\chi_\phi = \frac{1}{r}\left[-n\beta + \alpha\sin\psi\right] &= \frac{1}{r}\left[-\frac{n^2}{r}(q_5 + q_6 s + q_7 s^2 + q_8 s^3) - \frac{n\cos\psi}{r}(q_1 + q_2 s)\right.\\
&\qquad \left. - n(q_9 + q_{10}s) + \sin\psi(q_6 + 2q_7 s + 3q_8 s^2)\right] - \frac{\sin\psi}{r}\gamma_s\\
&= q_1\left(-\frac{n\cos\psi}{r^2}\right) + q_2\left(-\frac{ns\cos\psi}{r^2}\right) + q_5\left(-\frac{n^2}{r^2}\right) + q_6\left(\frac{\sin\psi}{r} - \frac{sn^2}{r^2}\right) + q_7\left(\frac{2s\sin\psi}{r} - \frac{n^2s^2}{r^2}\right)\\
&\quad + q_8\left(\frac{3s^2\sin\psi}{r} - \frac{n^2s^3}{r^2}\right) + q_9\left(-\frac{n}{r}\right) + q_{10}\left(-\frac{ns}{r}\right) - \frac{\sin\psi}{r}\gamma_s \qquad . \qquad (67)
\end{aligned}$$

From Equations 54, 55 and 56

$$\begin{aligned}
\chi_{s\phi} = {} & q_1\left(\frac{\sin\psi\cos\psi}{2r^2}\right) + q_2\left(\frac{s\sin\psi\cos\psi}{2r^2} - \frac{3\cos\psi}{2r}\right) + q_3\left(-\frac{n\cos\psi}{2r^2}\right) + q_4\left(-\frac{ns\cos\psi}{2r^2}\right)\\
& + q_5\left(\frac{n\sin\psi}{r^2}\right) + q_6\left(\frac{ns\sin\psi}{r^2} - \frac{2n}{r}\right) + q_7\left(\frac{ns^2\sin\psi}{r^2} - \frac{4ns}{r}\right) + q_8\left(\frac{ns^3\sin\psi}{r^2} - \frac{6ns^2}{r}\right)\\
& + q_9\left(\frac{\sin\psi}{r}\right) + q_{10}\left(-1 + \frac{s}{r}\sin\psi\right) + \frac{n}{r}\gamma_s \qquad , \qquad (68)
\end{aligned}$$

so that, in matrix form,

ORIGINAL PAGE IS OF POOR QUALITY

$$[H_{\chi q}]^T = \begin{array}{c|c|c|c}
\chi_s & \chi_\phi & \chi_{s\phi} & \\
0 & \frac{-n\cos\psi}{r^2} - \frac{\sin\psi}{r} H_{11}{}^{\gamma q} & \frac{\sin\psi\cos\psi}{2r^2} + \frac{n}{r} H_{11}{}^{\gamma q} & 1 \\
0 & \frac{-ns\cos\psi}{r^2} - \frac{\sin\psi}{r} H_{12}{}^{\gamma q} & \frac{1}{2}\left(\frac{s\sin\psi\cos\psi}{r^2} - \frac{3\cos\psi}{r}\right) + \frac{n}{r} H_{12}{}^{\gamma q} & 2 \\
0 & \frac{-\sin\psi}{r} H_{13}{}^{\gamma q} & \frac{-n\cos\psi}{2r^2} + \frac{n}{r} H_{13}{}^{\gamma q} & 3 \\
0 & \frac{-\sin\psi}{r} H_{14}{}^{\gamma q} & \frac{-ns\cos\psi}{2r^2} + \frac{n}{r} H_{14}{}^{\gamma q} & 4 \\
0 & \frac{-n^2}{r^2} - \frac{\sin\psi}{r} H_{15}{}^{\gamma q} & \frac{n\sin\psi}{r^2} + \frac{n}{r} H_{15}{}^{\gamma q} & 5 \\
0 & \frac{\sin\psi}{r} - \frac{n^2 s}{r^2} - \frac{\sin\psi}{r} H_{16}{}^{\gamma q} & \frac{ns\sin\psi}{r^2} - \frac{2n}{r} + \frac{n}{r} H_{16}{}^{\gamma q} & 6 \\
2 & \frac{2s\sin\psi}{r} - \frac{n^2 s^2}{r^2} - \frac{\sin\psi}{r} H_{17}{}^{\gamma q} & \frac{ns^2\sin\psi}{r^2} - \frac{4ns}{r} + \frac{n}{r} H_{17}{}^{\gamma q} & 7 \\
6s & \frac{3s^2\sin\psi}{r} - \frac{n^2 s^3}{r^2} - \frac{\sin\psi}{r} H_{18}{}^{\gamma q} & \frac{ns^3\sin\psi}{r^2} - \frac{6ns^2}{r} + \frac{n}{r} H_{18}{}^{\gamma q} & 8 \\
0 & -\frac{n}{r} - \frac{\sin\psi}{r} H_{19}{}^{\gamma q} & \frac{\sin\psi}{r} + \frac{n}{r} H_{19}{}^{\gamma q} & 9 \\
0 & -\frac{ns}{r} - \frac{\sin\psi}{r} H_{1,10}{}^{\gamma q} & -1 + \frac{s}{r}\sin\psi + \frac{n}{r} H_{1,10}{}^{\gamma q} & 10
\end{array} \qquad (69)$$

where

$$\{H_{\chi\gamma}\} = \left\{\begin{array}{c} 0 \\ \frac{-\sin\psi}{r} \\ \frac{n}{r} \end{array}\right\} , \qquad (70)$$

has been used to form the terms involving $H_{ij}{}^{\gamma q}$.

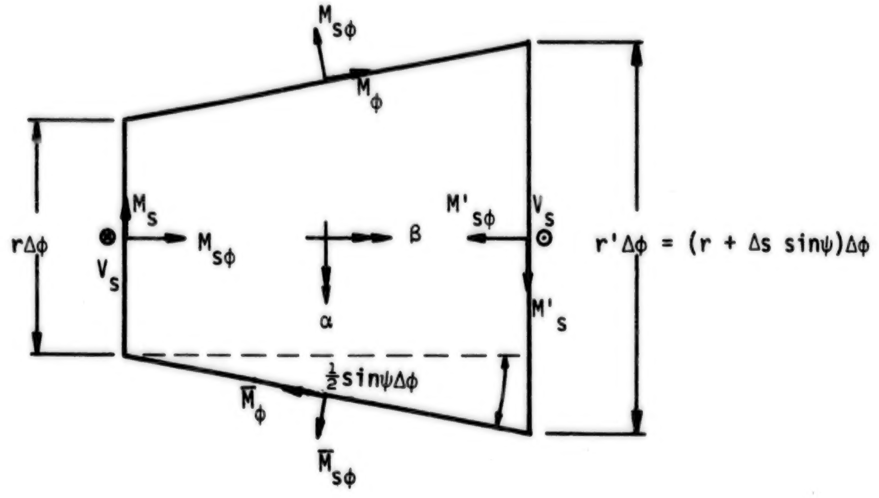
5.9.5.5 Development of $[H_{\gamma q}]$

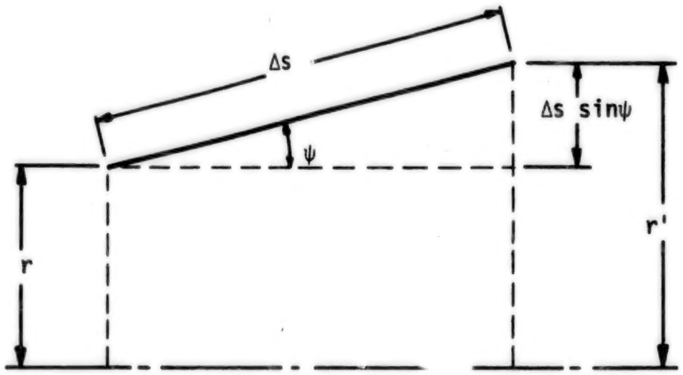
The first row of $[H_{\gamma q}]$ is $\lfloor H_{\gamma_s q} \rfloor$. The second row of $[H_{\gamma q}]$ as obtained from Equation 33, has zeroes for the first eight elements and for the remaining two:

$$H_{29}{}^{\gamma q} = 1; \quad H_{2,10}{}^{\gamma q} = s \quad .$$

$\lfloor H_{\gamma_s q} \rfloor$ is developed by means of an equation of moment equilibrium into which the meridional component of transverse shear enters.

Consider a surface element as shown below. The symbols refer to physical quantities rather than to their Fourier components.


$M_{s\phi}$
M_ϕ
$r\Delta\phi$
M_s
$M_{s\phi}$
V_s
β
α
$M'_{s\phi}$
V_s
M'_s
$r'\Delta\phi = (r + \Delta s \sin\psi)\Delta\phi$
$\frac{1}{2}\sin\psi\Delta\phi$
$\overline{M}_\phi$
$\overline{M}_{s\phi}$


Δs
$\Delta s \sin\psi$
ψ
r
r'

The positive direction of moments shown above is consistent with the positive convention for curvatures.

Equilibrium about the α-axis requires, considering terms of the same order of smallness, that

$$V_s r\Delta\phi\Delta s + (M_s' r'\Delta\phi - M_s r\Delta\phi) + (\overline{M}_{s\phi} - M_{s\phi})\Delta s - M_\phi \sin\psi\Delta\phi\Delta s = 0 \ . \tag{71}$$

To first order in Δs and $\Delta\phi$,

$$M_s' r' = rM_s + \frac{\partial}{\partial s}(rM_s)\Delta s \quad , \tag{72}$$

and

$$\overline{M}_{s\phi} = M_{s\phi} + \frac{\partial}{\partial\phi}(M_{s\phi})\Delta\phi \quad . \tag{73}$$

Substituting Equation 72 and 73 into Equation 71, and dividing by $\Delta\phi\Delta s$,

$$rV_s + \frac{\partial}{\partial s}(rM_s) + \frac{\partial}{\partial\phi}(M_{s\phi}) - M_\phi \sin\psi = 0 \ . \tag{74}$$

Since $M_{s\phi} = M_{s\phi_n}\sin(n\phi)$ for the condition described by Equation 46, we may write,

$$rV_{s_n} + \frac{\partial}{\partial s}(rM_{s_n}) + nM_{s\phi_n} - M_{\phi_n}\sin\psi = 0 \ . \tag{75}$$

Since $\gamma_{s_n} = \frac{1}{t_s G_{11}} V_{s_n}$, we can now write, suppressing n as a subscript:

$$\gamma_s = \frac{1}{rt_s G_{11}}\left[- \frac{\partial}{\partial s}(rM_s) - nM_{s\phi} + M_\phi \sin\psi\right], \tag{76}$$

and since

$$\begin{Bmatrix} M_s \\ M_\phi \\ M_{s\phi} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & 0 \\ D_{12} & D_{22} & 0 \\ 0 & 0 & D_{33} \end{bmatrix} \begin{Bmatrix} \chi_s \\ \chi_\phi \\ \chi_{s\phi} \end{Bmatrix} \quad ,$$

we arrive at

$$\gamma_s = \frac{1}{rt_s G_{11}}\left[- \frac{\partial}{\partial s}\Big(r(D_{11}\chi_s + D_{12}\chi_\phi)\Big) - nD_{33}\chi_{s\phi} + \sin\psi(D_{12}\chi_s + D_{22}\chi_\phi)\right] . \tag{77}$$

Define

$$\{\bar{\chi}\} = \begin{Bmatrix} \bar{\chi}_s \\ \bar{\chi}_\phi \\ \bar{\chi}_{s\phi} \end{Bmatrix} = \begin{Bmatrix} \chi_s \\ \chi_\phi \\ \chi_{s\phi} \end{Bmatrix} - \{H_{\chi\gamma}\}\gamma_s = \begin{Bmatrix} \chi_s \\ \chi_\phi \\ \chi_{s\phi} \end{Bmatrix} - \begin{Bmatrix} 0 \\ \frac{-\sin\psi}{r} \\ \frac{n}{r} \end{Bmatrix} \gamma_s \quad . \tag{78}$$

The relationship between $\{\bar{\chi}\}$ and the independent constants has already been found (Equation 69). Substitute Equation 78 into 77.

$$\gamma_s = \frac{1}{r t_s G_{11}} \left[-\frac{\partial}{\partial s}\left(r(D_{11}\bar{\chi}_s + D_{12}\bar{\chi}_\phi) \right) - nD_{33}\bar{\chi}_{s\phi} + \sin\psi(D_{12}\bar{\chi}_s + D_{22}\bar{\chi}_\phi) \right.$$
$$\left. -\left(\frac{n^2 D_{33} + \sin^2\psi D_{22}}{r} \right) \gamma_s + D_{12}\sin\psi \frac{\partial \gamma_s}{\partial s} \right] \quad . \tag{79}$$

Equation 79 is an exact relationship that indicates γ_s to be a variable over the surface. In order to comply with an earlier assumption, Equation 33, γ_s must be assumed constant in energy calculations. Let $\bar{\gamma}_s$ be the weighted average value of γ_s obtained by integrating Equation 79 over the surface.

$$\bar{\gamma}_s = \frac{1}{\ell r_{av}} \int_0^\ell r\gamma_s ds \quad , \tag{80}$$

where

$$r_{av} = \frac{1}{2}(r_a + r_b) \quad .$$

Thus, assuming γ_s to be constant on the right hand side of Equation 79

$$\bar{\gamma}_s = \frac{1}{\ell t_s G_{11} r_{av}} \left[r_a \left(D_{11}\bar{\chi}_s(a) + D_{12}\bar{\chi}_\phi(a) \right) - r_b \left(D_{11}\bar{\chi}_s(b) + D_{12}\bar{\chi}_\phi(b) \right) \right.$$
$$\left. + \int_0^\ell \left(-nD_{33}\bar{\chi}_{s\phi} + \sin\psi(D_{12}\bar{\chi}_s + D_{22}\bar{\chi}_\phi) \right) ds \right] - \frac{I_{02}}{\pi} \left(\frac{n^2 D_{33} + \sin^2\psi D_{22}}{\ell t_s G_{11} r_{av}} \right) \bar{\gamma}_s \quad , \tag{81}$$

where

$$I_{mn} = \pi \int_0^\ell s^m r^{1-n} ds \quad . \tag{82}$$

Write

$$Q = \ell t_s G_{11} r_{av} \left[1 + \frac{I_{02}}{\pi} \left(\frac{n^2 D_{33} + \sin^2\psi D_{22}}{\ell t_s G_{11} r_{av}} \right) \right] . \tag{83}$$

Then

$$\begin{aligned} \bar{\gamma}_s = \frac{1}{Q} \Big[& r_a \left(D_{11}\bar{\chi}_s(a) + D_{12}\bar{\chi}_\phi(a) \right) - r_b \left(D_{11}\bar{\chi}_s(b) + D_{12}\bar{\chi}_\phi(b) \right) \\ & + \int_0^\ell \left(- nD_{33}\bar{\chi}_{s\phi} + \sin\psi(D_{12}\bar{\chi}_s + D_{22}\bar{\chi}_\phi) \right) ds \Big] . \end{aligned} \tag{84}$$

Evaluation of the right side of Equation 84 by substituting for the $\bar{\chi}$'s from Equation 69, now gives γ_s in the form $\gamma_s = \lfloor H_{\gamma_s q} \rfloor \{q\}$; $\lfloor H_{\gamma_s q} \rfloor$ is the first row of $[H_{\gamma q}]$. The nonzero elements of the second row of $[H_{\gamma q}]$ are:

$$H_{29}{}^{\gamma q} = 1; \qquad H_{2,10}{}^{\gamma q} = s.$$

On evaluation of the right hand side of Equation 84, we get

$$\left. \begin{aligned} H_{11}{}^{\gamma q} &= \frac{1}{Q} \left[D_{12} n\cos\psi \left(\frac{1}{r_b} - \frac{1}{r_a} \right) - \frac{n}{2} \cos\psi\sin\psi \, \frac{I_{03}}{\pi} \left(D_{33} + 2D_{22} \right) \right] , \\ H_{12}{}^{\gamma q} &= \frac{1}{Q} \left[D_{12} \, \frac{n\ell\cos\psi}{r_b} - \frac{1}{2} \, n\sin\psi\cos\psi \, \frac{I_{13}}{\pi} \left(3D_{33} + D_{22} \right) + \frac{3}{2} \, nD_{33}\cos\psi \, \frac{I_{02}}{\pi} \right] , \\ H_{13}{}^{\gamma q} &= \frac{1}{Q} \left[\frac{1}{2} \, n^2 D_{33} \cos\psi \frac{I_{03}}{\pi} \right] , \\ H_{14}{}^{\gamma q} &= \frac{1}{Q} \left[\frac{1}{2} \, n^2 D_{33} \cos\psi \frac{I_{13}}{\pi} \right] , \\ H_{15}{}^{\gamma q} &= \frac{1}{Q} \left[n^2 D_{12} \left(\frac{1}{r_b} - \frac{1}{r_a} \right) - n^2 \sin\psi \, \frac{I_{03}}{\pi} \left(2D_{33} + D_{22} \right) \right] , \\ H_{16}{}^{\gamma q} &= \frac{1}{Q} \left[D_{12} \left(\frac{n^2\ell}{r_b} \right) - n^2 \sin\psi \, \frac{I_{13}}{\pi} \left(2D_{33} + D_{22} \right) + \frac{I_{02}}{\pi} \left(2n^2 D_{33} + \sin^2\psi D_{22} \right) \right] , \\ H_{17}{}^{\gamma q} &= \frac{1}{Q} \Big[2D_{11}(r_a - r_b) + D_{12} \left(\frac{n^2\ell^2}{r_b} \right) + \frac{2I_{12}}{\pi} \left(2n^2 D_{33} + \sin^2\psi D_{22} \right) \\ &\qquad - n^2 \sin\psi \frac{I_{23}}{\pi} \left(2D_{33} + D_{22} \right) \Big] , \end{aligned} \right\} \tag{85a}$$

$$\left.\begin{aligned}
H_{18}{}^{\gamma q} &= \frac{1}{Q}\left[-D_{11}6\ell r_b + D_{12}\left(\frac{n^2\ell^3}{r_b}\right) + \frac{3I_{22}}{\pi}\left(2n^2D_{33} + \sin^2\psi D_{22}\right)\right. \\
&\qquad \left. - n^2\sin\psi\,\frac{I_{33}}{\pi}\left(2D_{33} + D_{22}\right)\right] \quad , \\
H_{19}{}^{\gamma q} &= \frac{1}{Q}\left[-\, n\sin\psi\frac{I_{02}}{\pi}\left(D_{22} + D_{33}\right)\right] , \\
H_{1,10}{}^{\gamma q} &= \frac{1}{Q}\left[n\ell\left(D_{12} + D_{33}\right) - n\sin\psi\,\frac{I_{12}}{\pi}\left(D_{22} + D_{33}\right)\right] .
\end{aligned}\right\} \tag{85b}$$

We are now able to compute $[H_{uq}]$ from Equation 59 and to evaluate the constants $H_{ij}{}^{\gamma q}$ in Equation 69.

5.9.5.6 Explicit Form for $[K^q]$

$[K^q]$ is, for convenience, separated into parts due to stretching transverse shear, and bending.

$$[K^q] = [K^{q\varepsilon}] + [K^{q\gamma}] + [K^{q\chi}] \quad , \tag{86}$$

where

$$[K^{q\varepsilon}] = \pi t \int_0^\ell [H_{\varepsilon q}]^T[E][H_{\varepsilon q}]\, r ds \quad , \tag{87}$$

$$[K^{q\gamma}] = \pi t_s \int_0^\ell [H_{\gamma q}]^T[G][H_{\gamma q}]\, r ds \; , \tag{88}$$

$$[K^{q\chi}] = \pi \int_0^\ell [H_{\chi q}]^T[D][H_{\chi q}]\, r ds \quad . \tag{89}$$

In the case of $[K^{q\gamma}]$, the calculation is performed as follows. Let

$$[H_{\gamma q}] = \begin{bmatrix} H_{\gamma_s q} \\ \hline H_{\gamma_\phi q} \end{bmatrix} \quad . \tag{90}$$

Then, since $[G]$ is a diagonal matrix and $\{H_{\gamma_s q}\}$ is independent of s, we may write

$$[K^{q\gamma}] = [K^{q\gamma_s}] + [K^{q\gamma_\phi}] \quad , \tag{91}$$

where, noting that I_{00} is defined in Equation 82,

$$[K^{q\gamma_s}] = t_s G_{11} I_{00} \{H_{\gamma_s q}\}^T \{H_{\gamma_s q}\} \quad , \tag{92}$$

and

$$[K^{q\gamma_\phi}] = \pi t_s G_{22} \int_0^{\ell} \{H_{\gamma_\phi q}\}^T \{H_{\gamma_\phi q}\} \, rds \quad . \tag{93}$$

$[K^{q\gamma_s}]$ is a full symmetric ten by ten matrix obtained by crossmultiplying the terms in Equation 79. Explicitly:

$$K_{ij}^{q\gamma_s} = \left(H_{1i}^{\gamma q} \, H_{1j}^{\gamma q} \right) t_s G_{11} I_{00} \quad . \tag{94}$$

$[K^{q\gamma_\phi}]$ is a symmetric matrix that is zero except in the 9th and 10th rows and columns. These terms are:

$$\left. \begin{aligned} K_{99}^{q\gamma_\phi} &= t_s G_{22} I_{00} \quad , \\ K_{9,10}^{q\gamma_\phi} = K_{10,9}^{q\gamma_\phi} &= t_s G_{22} I_{10} \quad , \\ K_{10,10}^{q\gamma_\phi} &= t_s G_{22} I_{20} \quad . \end{aligned} \right\} \tag{95}$$

In the case of $[K^{q\varepsilon}]$, explicit formulas were written for the individual terms. The terms include integrals over the slant length in the form given by Equation 82. The formulas which are not difficult to write, will be omitted in the interest of brevity.

In the case of $[K^{q\chi}]$ explicit formulas were also written for the individual terms. The formulas occupied eleven typewritten pages and it was virtually impossible to verify their correctness, not to mention that of the corresponding FORTRAN code. At this point it was decided that the computer should be taught to perform analysis, i.e. that it should evaluate the integral in Equation 89 from the formulas for the elements of $[H_{\chi q}]$ that include the variable of integration (see Equation 69). The analysis procedure involves recognition of the fact that the integral of the product of terms A and B where

$$A = \frac{as^m}{r^n} \qquad \text{and} \qquad B = \frac{bs^i}{r^j}$$

is

$$\pi \int_0^{\ell} ABrds = abI_{m+i,n+j} \quad . \tag{96}$$

A subroutine was written in which the powers of r and s in the individual products of $[H_{\chi q}]^T[D][H_{\chi q}]$ were logically added and used to call the appropriate integrals. The programmer's task was greatly reduced, and the reliability was correspondingly increased, because only about sixty coefficients (see Equation 69) were involved rather than about six hundred. The only use made of the explicit formulas for the elements of $[K^{q\chi}]$ was to check the results of the computer subroutine for enough terms to ensure its correctness.

5.9.5.7 Summary of Procedures

In summary, the computational steps required to form the stiffness matrix of the conical shell element for the general case, $n > 0$ with finite shear flexibility, for both the starred and unstarred quantities are

1. Form $[\overline{H}_{uq}]$ and $\{H_{u\gamma}\}$. See Section 5.9.5.3.
2. Form $\{H_{\gamma_s q}\}$ from Equation 85.
3. Form $[H_{uq}]$ from Equation 59.
4. Invert $[H_{uq}]$.
5. Form $[K^{q\varepsilon}]$. See Section 5.9.5.6.
6. Form $[K^{q\gamma}]$. See Section 5.9.5.6.
7. Form $[K^{q\chi}]$. See Section 5.9.5.6
8. Form $[K_n{}^q]$ from Equation 86.
9. Form $[K_n{}^u]$ from Equation 29.

5.9.5.8 Evaluation of Integral, I_{mn}

The integral to be evaluated is

$$I_{mn} = \pi \int_0^{\ell} s^m r^{1-n} \, ds \quad , \qquad (96)$$

where

$$r = a + bs, \quad a = r_a, \quad \text{and} \quad b = \frac{r_b - r_a}{\ell} \quad .$$

Dwight's Table of Integrals, Formula 89, gives an explicit formula for the evaluation of I_{mn}. In terms of our notation the formula is

$$I_{mn} = \pi \frac{m!}{b^{m+1}} \left[\sum_{\substack{j=0\\ j\neq m-n+2}}^{m} \frac{(-a)^{j} r^{m-n-j+2}}{(m-j)!\; j!\; (m-n-j+2)} + \underset{j=m-n+2}{\frac{(-a)^{m-n+2}\log(r)}{(m-n+2)!\;(n-2)!}} \right]_{r_a}^{r_b} . \tag{97}$$

The second (log r) term replaces the term in the series for $j = m-n+2$ and should not be included unless $j = m-n+2$ for some j. This will happen if, and only if, $m + 2 \geq n \geq 2$.

5.9.6 Stiffness Matrix for Finite Transverse Shear Flexibility, n = 0

The formulas presented in the preceding section are valid for $n = 0$, provided that the final stiffness matrix, as computed from the formulas of the preceding section, is multiplied by two.

5.9.7 Stiffness Matrix for Zero Transverse Shear Flexibility, n > 0

The first consequence of zero transverse shear flexibility is that β_{na} and β_{nb} disappear as independent degrees of freedom. The following equations of constraint then occur:

$$\beta_{na} = \frac{1}{r_a}\left(nw_{na} + \cos\psi u_{na}\right) , \tag{98}$$

$$\beta_{nb} = \frac{1}{r_b}\left(nw_{nb} + \cos\psi u_{nb}\right) . \tag{99}$$

In the element stiffness matrix, $[K_n^u]$, the fifth and tenth rows and columns are deleted. $[K_n^u]$ is evaluated as before with the following changes:

(a.) In the transformation matrix, $[H_{uq}]$, the fifth and tenth rows and the ninth and tenth columns are deleted. Also $[H_{uq}] = [\overline{H}_{uq}]$.

(b.) In the stiffness matrix referred to independent constants, $[K_n^q]$, the ninth and tenth rows and columns are deleted. Also $[K^{q\gamma}] = 0$, and all $H^{\gamma q}$ terms in $[K^{q\chi}]$ are set equal to zero.

Note that it is not correct simply to delete the fifth and tenth rows and columns of $[K_n^u]$ from the results of Section 5.9.5.5.

5.9.8 Stiffness Matrix for Zero Transverse Shear Flexibility, n = 0

The formulas obtained by the procedures outlined in the preceding section are valid provided that the final stiffness matrix is multiplied by two.

5.9.9 Static Loads

Special procedures are required in generating loads and reducing data for axisymmetric shells due to the use of harmonic functions of azimuth position in the analysis.

The basic coordinate system for the shell is a cylindrical (ϕ,z,r) system, as shown in Figure 3 below. The local coordinate systems used to define loads and displacement components at grid points may either be cylindrical (ϕ,z,r) or spherical (ϕ,θ,ρ). All such coordinate systems must have the same azimuth reference as the basic coordinate system.

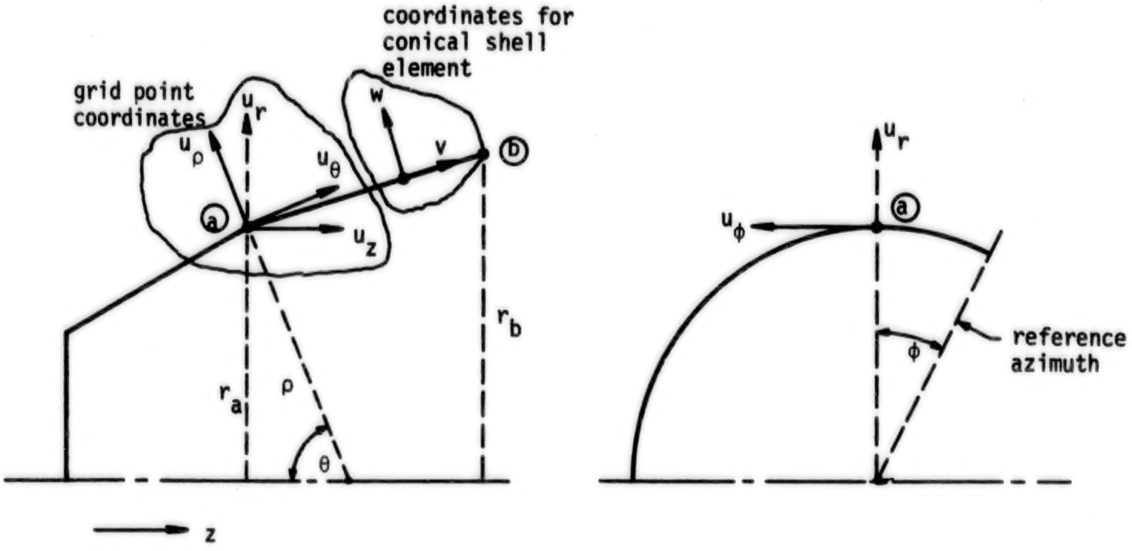

coordinates for
conical shell
element
grid point
coordinates
u_r
w
u_ρ
u_θ
v
ⓑ
ⓐ
u_z
r_b
r_a
ρ
θ
z
u_r
u_ϕ
ⓐ
ϕ
reference
azimuth

Figure 3. Coordinate systems.

The element coordinate system is a conical system (ϕ,s,w) where s is a parallel to the generator of the element.

The motions along each grid circle are the components of the vector,

$$\{u_g\} = \lfloor u_\phi, u_z, u_r, \theta_\phi, \theta_z, \theta_r \rfloor^T \quad . \tag{100}$$

The components of $\{u_g\}$ depend on azimuth position. Such dependence is eliminated by Fourier series expansion as explained in Section 5.9.2.

Under certain conditions either θ_z or θ_r, or both, will be eliminated as degrees of freedom by single-point constraints. For example, if all of the elements of the structure are rigid in transverse shear both θ_z and θ_r will be eliminated. Also if the elements joined together at a grid circle have the same cone angle, either θ_z or θ_r should be eliminated.

The degrees of freedom used in analysis are the Fourier coefficients of the motions at grid points. The loads, or generalized forces, acting on the degrees of freedom are computed by the following fundamental theorem.

Let f_i be the forces applied to points where the motions are y_i. Let the motions y_i be linearly dependent on a set of parameters u_j. Then the generalized force on each u_j is

$$P_j = \sum_i f_i \frac{\partial y_i}{\partial u_j} = \sum_i H_{ij} f_i \quad . \tag{101}$$

In the present instance the motions y_i are the components of the $\{u_g\}$ vector defined by Equation 100, and the parameters u_j are the harmonic coefficients in the Fourier expansion of $\{u_g\}$. The expansion of $\{u_g\}$ is explicitly:

$$\left.\begin{aligned}
u_\phi &= \sum_{n=1}^{m} u_{\phi n}\sin n\phi + u_{\phi o}{}^{*} - \sum_{n=1}^{m} u_{\phi n}{}^{*}\cos n\phi \quad , \\
u_z &= \sum_{n=1}^{m} u_{zn}\cos n\phi + u_{zo} + \sum_{n=1}^{m} u_{zn}{}^{*}\sin n\phi \quad , \\
u_r &= \sum_{n=1}^{m} u_{rn}\cos n\phi + u_{ro} + \sum_{n=1}^{m} u_{rn}{}^{*}\sin n\phi \quad , \\
\theta_\phi &= \sum_{n=1}^{m} \theta_{\phi n}\cos n\phi + \theta_{\phi o} + \sum_{n=1}^{m} \theta_{\phi n}{}^{*}\sin n\phi \quad , \\
\theta_z &= \sum_{n=1}^{m} \theta_{zn}\sin n\phi + \theta_{zo}{}^{*} - \sum_{n=1}^{m} \theta_{zn}{}^{*}\cos n\phi \quad , \\
\theta_r &= \sum_{n=1}^{m} \theta_{rn}\sin n\phi + \theta_{ro}{}^{*} - \sum_{n=1}^{m} \theta_{rn}{}^{*}\cos n\phi \quad .
\end{aligned}\right\} \tag{102}$$

ORIGINAL PAGE IS OF POOR QUALITY

The motions corresponding to different harmonic orders are uncoupled. Also the motions corresponding to starred and unstarred coefficients are uncoupled. The minus signs have been introduced in order to make the stiffness matrices for starred and unstarred parameters identically the same for n > 0.

The load vectors (generalized forces) acting on the degrees of freedom are designated by:

$$\left.\begin{aligned}
\{P_o\}^T &= \lfloor P\;\;, P_{ro}, M_{\phi o} \rfloor \;, \\
\{P_o^*\}^T &= \lfloor P_{\phi o}^*, M_{zo}^*, M_{ro}^* \rfloor \;, \\
\{P_n\}^T &= \lfloor P_{\phi n}, P_{zn}, P_{rn}, M_{\phi n}, M_{zn}, M_{rn} \rfloor, \; n > 0 \\
\{P_n^*\}^T &= \lfloor P_{\phi n}^*, P_{zn}^*, P_{rn}^*, M_{\phi n}^*, M_{zn}^*, M_{rn}^* \rfloor, \; n > 0 \quad .
\end{aligned}\right\} \qquad (103)$$

$\{P_o\}$ and $\{P_o^*\}$ are combined into a single load vector. $\{P_n\}$ and $\{P_n^*\}$ may be regarded as separate loading conditions for the same idealized structure, since the stiffness matrices for the starred and unstarred systems are identical for n > 0.

The specific treatment for various types of loading are discussed below.

5.9.9.1 Loads Designated at Grid Points

The following options are available to the user for the specification of static loads applied directly to the grid points of an axisymmetric shell.

(a) Specification of the harmonic coefficients of a line load density, $\{f\}$, along a grid circle. (The local coordinate system need not be cylindrical.)

(b) Specification of concentrated loads $\{F_i^c\}$ at azimuth positions ϕ_i.

(c) Specification of uniformly distributed line loads $\{f_j^d\}$ between azimuth positions, $\phi_{1j} < \phi_j < \phi_{2j}$.

Considering option (a), let the component of the line load density in the ϕ direction be given by

$$f_\phi(\phi) = a_{\phi o} + \sum_{n=1}^{m} a_{\phi n}\cos n\phi + \sum_{n=1}^{m} b_{\phi n}\sin n\phi \quad . \qquad (104)$$

Then from Equations 101 and 102, replacing the summation by an integration,

$$\left.\begin{aligned}
P_{\phi o}^{*} &= \int_{o}^{2\pi} f_{\phi}(\phi)\,\frac{\partial u_{\phi}}{\partial u_{\phi o}^{*}}\,r d\phi = 2\pi r a_{\phi o} \quad , \\
P_{\phi n} &= \int_{o}^{2\pi} f_{\phi}(\phi)\,\frac{\partial u_{\phi}}{\partial u_{\phi n}}\,r d\phi = \pi r b_{\phi n}, \quad n > 0 \ , \\
P_{\phi n}^{*} &= \int_{o}^{2\pi} f_{\phi}(\phi)\,\frac{\partial u_{\phi}}{\partial u_{\phi n}^{*}}\,r d\phi = -\pi r a_{\phi n}, \quad n > 0 \ .
\end{aligned}\right\} \qquad (105)$$

Let the component of load in the z direction be given by

$$f_z(\phi) = a_{zo} + \sum_{n=1}^{m} a_{zn}\cos n\phi + \sum_{n=1}^{m} b_{zn}\sin n\phi \quad . \qquad (106)$$

Then from Equations 101 and 102

$$\left.\begin{aligned}
P_{zo} &= 2\pi r a_{zo} \quad , \\
P_{zn} &= \pi r a_{zn}, \quad n > 0 \ , \\
P_{zn}^{*} &= \pi r b_{zn}, \quad n > 0 \ .
\end{aligned}\right\} \qquad (107)$$

The other components of generalized force and moment follow either the pattern of Equation 105 or the pattern of Equation 107.

Considering option (b), let the component of the ith concentrated load in the ϕ direction be $F_{\phi i}{}^{c}$. Then from Equations 101 and 102

$$\left.\begin{aligned}
P_{\phi o}^{*} &= \sum_{i} F_{\phi i}{}^{c} \quad , \\
P_{\phi n} &= \sum_{i} F_{\phi i}{}^{c} \sin n\phi_{i} \ , \\
P_{\phi n}^{*} &= -\sum_{i} F_{\phi i}{}^{c} \cos n\phi_{i} .
\end{aligned}\right\} \qquad (108)$$

For the components of concentrated load in the z direction, $F_{zi}{}^{c}$,

$$\left.\begin{aligned} P_{zo} &= \sum_i F_{zi}{}^{c} \quad , \\ P_{zn} &= \sum_i F_{zi}{}^{c} \cos n\phi_i \quad , \\ P_{zn}{}^{*} &= \sum_i F_{zi}{}^{c} \sin n\phi_i \quad . \end{aligned}\right\} \qquad (109)$$

and similarly for the other components of force and moment.

Considering option (c), let the component of the jth distributed load in the ϕ direction be $f_{\phi j}{}^{d}$. Then from Equations 101 and 102

$$\left.\begin{aligned} P_{\phi o}{}^{*} &= \sum_j \int_{\phi_{1j}}^{\phi_{2j}} f_{\phi j}{}^{d} r\,d\phi = r \sum_j f_{\phi j}{}^{d}(\phi_{2j} - \phi_{1j}) \quad , \\ P_{\phi n} &= \sum_j \int_{\phi_{1j}}^{\phi_{2j}} f_{\phi j}{}^{d} \sin n\phi \; r\,d\phi = -\frac{r}{n} \sum_j f_{\phi j}{}^{d}(\cos n\phi_{2j} - \cos n\phi_{1j}) \quad , \\ P_{\phi n}{}^{*} &= -\sum_j \int_{\phi_{1j}}^{\phi_{2j}} f_{\phi j}{}^{d} \cos n\phi \; r\,d\phi = -\frac{r}{n} \sum_j f_{\phi j}{}^{d}(\sin n\phi_{2j} - \sin n\phi_{1j}) \quad , \end{aligned}\right\} \qquad (110)$$

and for the components of load in the z direction.

$$\left.\begin{aligned} P_{zo} &= r \sum_j f_{zj}{}^{d}(\phi_{2j} - \phi_{1j}) \quad , \\ P_{zn} &= \frac{r}{n} \sum_j f_{zj}{}^{d}(\sin n\phi_{2j} - \sin n\phi_{1j}) \quad , \\ P_{zn}{}^{*} &= -\frac{r}{n} \sum_j f_{zj}{}^{d}(\cos n\phi_{2j} - \cos n\phi_{1j}) \quad . \end{aligned}\right\} \qquad (111)$$

5.9.9.2 Pressure Loads

The user has the option to specify pressure loads acting normal to the surface of a conical shell element. The direction of the load is in the w direction shown in Figure 3. Each pressure load, p_j, (there may be several j's for each element) is uniform over the slant length of the element, ℓ, and between azimuth positions $\phi_{1j} < \phi_j < \phi_{2j}$.

The pressure load is beamed to the adjacent grid points in such manner that the center of pressure is preserved. The generalized forces at grid point ⓐ due to pressure load on the conical shell element between grid points ⓐ and ⓑ in Figure 3 are

$$\left.\begin{aligned} P_{w_o} &= \ell\left(\frac{r_a}{3}+\frac{r_b}{6}\right)\sum_j p_j(\phi_{2j}-\phi_{1j}) \ , \\ P_{w_n} &= \frac{\ell}{n}\left(\frac{r_a}{3}+\frac{r_b}{6}\right)\sum_j p_j(\sin n\phi_{2j}-\sin n\phi_{1j}) \ , \\ P_{w_n}{}^{*} &= -\frac{\ell}{n}\left(\frac{r_a}{3}+\frac{r_b}{6}\right)\sum_j p_j(\cos n\phi_{2j}-\cos n\phi_{1j}) . \end{aligned}\right\} \tag{112}$$

The generalized forces in the w direction must, of course, be rotated into the directions of the degrees of freedom at the grid point.

5.9.9.3 Mass Distribution and Gravity Loads

The user specifies the total mass to be associated with a grid circle. The program then divides the total mass by 2 for $n > 0$ and constructs a diagonal 3 x 3 grid point mass matrix in the usual manner. (The total mass is used for $n = 0$.) The user can also specify c.g. offsets in the (r,z) plane and rotary inertias about three orthogonal axes. Rotary inertias are specified as inertia per unit length for each grid circle and are multiplied by $2\pi r$ for $n = 0$ and by πr for $n > 0$. The latter option (of specifying c.g. offsets and rotary inertias) has not been implemented in the program.

Structural mass density, ρ_s, and nonstructural mass density, ρ_{ns}, can also be specified for each conical shell element. The equivalent concentrated mass at grid point ⓐ due to the element between ⓐ and ⓑ in Figure 3 is

$$M_a = 2\pi\ell\left(\frac{r_a}{3}+\frac{r_b}{6}\right)[\rho_s t + \rho_{ns}] \tag{113}$$

which is divided by 2 for $n > 0$.

The gravity vector is specified by its magnitude, by its azimuth (ϕ_g) and by the angle (θ_g) that it makes with the negative z axis (see Figure 3). Gravity loads excite deformations in the $n = 0$ and $n = 1$ harmonics only. The distributed gravity forces acting on a grid circle are

$$\left.\begin{aligned} f_z^{\,g} &= -\frac{Mg}{2\pi r}\cos\theta_g \quad , \\ f_\phi^{\,g} &= -\frac{Mg}{2\pi r}\sin\theta_g\sin(\phi-\phi_g) \quad , \\ f_r^{\,g} &= \frac{Mg}{2\pi r}\sin\theta_g\cos(\phi-\phi_g) \quad , \end{aligned}\right\} \tag{114}$$

so that, using Equations 101 and 102

$$\left.\begin{aligned} P_{zo} &= \int_o^{2\pi} f_z^{\,g}\, r d\phi = -Mg\cos\theta_g \quad , \\ P_{\phi 1} &= \int_o^{2\pi} f_\phi^{\,g}\sin\phi\, r d\phi = -\frac{Mg}{2}\sin\theta_g\cos\phi_g \quad , \\ P_{\phi 1}^{*} &= -\int_o^{2\pi} f_\phi^{\,g}\cos\phi\, r d\phi = -\frac{Mg}{2}\sin\theta_g\sin\phi_g \quad , \\ P_{r1} &= \int_o^{2\pi} f_r^{\,g}\cos\phi\, r d\phi = \frac{Mg}{2}\sin\theta_g\cos\phi_g \quad , \\ P_{r1}^{*} &= \int_o^{2\pi} f_r^{\,g}\sin\phi\, r d\phi = \frac{Mg}{2}\sin\theta_g\sin\phi_g \quad , \end{aligned}\right\} \tag{115}$$

which are the only nonzero generalized forces. Note that the lateral load is divided equally between a radial (r) and a tangential (ϕ) component.

5.9.9.4 Loads Due to Temperature and Enforced Strains

Temperature is specified at each grid point circle in either of two ways:

(a) As the harmonic coefficients in the expansion

$$T = T_o + \sum_{n=1}^{m} T_n\cos n\phi + \sum_{n=1}^{m} T_n^{*}\sin n\phi \ . \tag{116}$$

This option has not been implemented in the program.

(b) As a set of values, T_i, at azimuth positions, ϕ_i. The temperature is assumed to vary linearly between azimuth positions where it is specified.

The harmonic coefficients in Equation 116 are in the correct form for use with the starred and unstarred degrees of freedom in Equation 102. If the second method of specifying temperature is used, the harmonic coefficients are computed as follows:

$$T_o = \frac{1}{2\pi}\int_o^{2\pi} T(\phi)d\phi = \frac{1}{4\pi}\sum_{i=1}^{k}(T_i + T_{i+1})(\phi_{i+1} - \phi_i) \quad , \tag{117}$$

where ϕ_k is the last specified position, $\phi_{k+1} = 2\pi + \phi_1$, and $T_{k+1} = T_1$.

$$\begin{aligned} T_n &= \frac{1}{\pi}\int_o^{2\pi} T(\phi)\cos n\phi \, d\phi = \frac{1}{\pi}\sum_{i=1}^{k}\int_{\phi_i}^{\phi_{i+1}}\left[T_i + \left(\frac{\phi - \phi_i}{\phi_{i+1} - \phi_i}\right)\left(T_{i+1} - T_i\right)\right]\cos n\phi \, d\phi \\ &= \frac{1}{\pi}\sum_{i=1}^{k}\frac{1}{\phi_{i+1} - \phi_i}\left[\left(\frac{T_i\phi_{i+1} - T_{i+1}\phi_i}{n}\right)\left(\sin n\phi_{i+1} - \sin n\phi_i\right)\right. \\ &\quad \left. + \left(\frac{T_{i+1} - T_i}{n^2}\right)\left(\cos n\phi_{i+1} - \cos n\phi_i + n\phi_{i+1}\sin n\phi_{i+1} - n\phi_i \sin n\phi_i\right)\right] . \end{aligned} \tag{118}$$

$$\begin{aligned} T_n^* &= \frac{1}{\pi}\int_o^{2\pi} T(\phi)\sin n\phi \, d\phi \\ &= \frac{1}{\pi}\sum_{i=1}^{k}\frac{1}{\phi_{i+1} - \phi_i}\left[\left(\frac{T_i\phi_{i+1} - T_{i+1}\phi_i}{n}\right)\left(-\cos n\phi_{i+1} + \cos n\phi_i\right)\right. \\ &\quad \left. + \frac{T_{i+1} - T_i}{n^2}\left(\sin n\phi_{i+1} - \sin n\phi_i - n\phi_{i+1}\cos n\phi_{i+1} + n\phi_i \cos n\phi_i\right)\right] \quad . \end{aligned} \tag{119}$$

The temperature is assumed to vary linearly along conical shell elements between grid circles. Thus for a conical shell element between grid circles ⓐ and ⓑ,

$$T = T^a + \frac{s}{\ell}(T^b - T^a) \quad . \tag{120}$$

The harmonic coefficients of temperature vary in the same manner. The incremental strains due to temperature and enforced strain are

$$\left.\begin{aligned} \Delta\varepsilon_s &= \alpha_s T + \delta\varepsilon_s \ , \\ \Delta\varepsilon_\phi &= \alpha_\phi T + \delta\varepsilon_\phi \ . \end{aligned}\right\} \tag{121}$$

The harmonic coefficients of strain have the same form, i.e.,

$$\left.\begin{aligned} \Delta\varepsilon_{so} &= \alpha_s T_o + \delta\varepsilon_{so} \quad , \\ \Delta\varepsilon_{sn} &= \alpha_s T_n + \delta\varepsilon_{sn} \quad , \\ \Delta\varepsilon_{sn}^{*} &= \alpha_s T_n^{*} + \delta\varepsilon_{sn}^{*} \quad , \end{aligned}\right\} \qquad (122)$$

and similarly for the ϕ components. Incremental shear strains, $\Delta\varepsilon_{s\phi}$, are excluded because they cannot be induced by temperature in an axisymmetric shell and because there is little application for enforced shear strains. The incremental strain vector for the nth unstarred parameter set may be written

$$\{\Delta\varepsilon_n\} = \left\{\begin{matrix} \Delta\varepsilon_{sn} \\ \Delta\varepsilon_{\phi n} \\ 0 \end{matrix}\right\} = \left\{\begin{matrix} \alpha_s T_n^{a} + \delta\varepsilon_{sn} \\ \alpha_\phi T_n^{a} + \delta\varepsilon_{\phi n} \\ 0 \end{matrix}\right\} + \frac{s}{\ell}\left(T_n^{b} - T_n^{a}\right)\left\{\begin{matrix} \alpha_s \\ \alpha_\phi \\ 0 \end{matrix}\right\} \quad . \qquad (123)$$

The generalized forces acting on strain components that are generated by the incremental strains are

$$\left.\begin{aligned} \{F_n\} &= \pi r t[E]\{\Delta\varepsilon_n\} \ , \quad n > 0 \quad , \\ \{F_o\} &= 2\pi r t[E]\{\Delta\varepsilon_o\} \quad , \end{aligned}\right\} \qquad (124)$$

and the generalized forces on the independent constants $\{q_n\}$ are

$$\begin{aligned} \{P_{qn}\} &= \int_0^{\ell} [H_{\varepsilon q}]^T \{F_n\}\, ds \\ &= \pi t \int_0^{\ell} [H_{\varepsilon q}]^T [E]\{\Delta\varepsilon_n\}\, r ds, \qquad n > 0 \quad . \end{aligned} \qquad (125)$$

The generalized forces acting on the degrees of freedom $\{u_{en}\}$ at the ends of the element and represented in the coordinate system for the element are

$$\{P_{en}\} = [H_{uq}^{-1}]\{P_{qn}\} \quad . \qquad (126)$$

The integration indicated in Equation 125 can be neatly expressed in terms of the integrals I_{mn} defined in Equation 82. The results for the unstarred parameter sets are, for $n > 0$,

$$P_{1n} = n(I_{01}A_n + I_{11}B_n) \quad ,$$

$$\{P_{2n}\} = n(I_{11}A_n + I_{21}B_n) \quad ,$$

$$\{P_{3n}\} = \sin\psi(I_{01}A_n + I_{11}B_n) \quad ,$$

$$\{P_{4n}\} = \sin\psi(I_{11}A_n + I_{21}B_n) + I_{01}C_n + I_{11}D_n \quad ,$$

$$\{P_{5n}\} = \cos\psi(I_{01}A_n + I_{11}B_n) \quad ,$$

$$\{P_{6n}\} = \cos\psi(I_{11}A_n + I_{21}B_n) \quad ,$$

$$\{P_{7n}\} = \cos\psi(I_{21}A_n + I_{31}B_n) \quad ,$$

$$\{P_{8n}\} = \cos\psi(I_{31}A_n + I_{41}B_n) \quad ,$$

$$\{P_{9n}\} = 0 \quad ,$$

$$\{P_{10,n}\} = 0 \quad ,$$

where

$$\left.\begin{aligned} A_n &= tE_{12}(\alpha_s T_n{}^a + \delta\varepsilon_{sn}) + tE_{22}(\alpha_\phi T_n{}^a + \delta\varepsilon_{\phi n}) \quad , \\ B_n &= \frac{t}{\ell}(T_n{}^b - T_n{}^a)(E_{12}\alpha_s + E_{22}\alpha_\phi) \quad , \\ C_n &= tE_{11}(\alpha_s T_n{}^a + \delta\varepsilon_{sn}) + tE_{12}(\alpha_\phi T_n{}^a + \delta\varepsilon_{\phi n}) \quad , \\ D_n &= \frac{t}{\ell}(T_n{}^b - T_n{}^a)(E_{11}\alpha_s + E_{12}\alpha_\phi) \quad . \end{aligned}\right\} \quad (127)$$

The results for starred parameters are the same with the substitution of starred temperature components. The results for $\{P_{qo}\}$ are obtained by multiplying the above formulas by 2.

5.9.9.5 Enforced Displacements

Enforced displacements at a grid ring are introduced into the program by specifying the harmonic coefficients in Equation 102. Note that the motions at all points on a grid ring in a given coordinate direction (e.g., u_ϕ) are thereby restrained. It is not possible to constrain motions at selected azimuth locations and to leave all other points free, without destroying the symmetry of the structure.

392

5.9.10 Data Reduction

Displacement information can be requested in two forms:

(a) The harmonic coefficients of motion on a grid circle defined in Equation 102.

(b) The motion at specified locations, ϕ_i, on a grid circle obtained by evaluating Equation 102 for each ϕ_i.

Internal force, moment and stress information can be requested in the same forms, evaluated at the two ends of a conical shell element (or within other types of elements if such are provided).

The internal force and moment components within a conical shell element are

$$\{F\} = \{F_s, F_\phi, F_{s\phi}\}^T \quad , \quad \text{membrane forces/unit width}$$

$$\{V\} = \{V_s, V_\phi\}^T \quad , \quad \text{transverse shear forces/unit width}$$

$$\{M\} = \{M_s, M_\phi, M_{s\phi}\}^T \quad , \quad \text{bending and twisting moments/unit width}$$

F_s, F_ϕ, V_s, M_s and M_ϕ have even symmetry, similar to u_z in Equation 102. $F_{s\phi}$, V_ϕ and $M_{s\phi}$ have odd symmetry, similar to u_ϕ in Equation 102.

The Fourier components of internal forces are evaluated by the following procedure, illustrated for unstarred components.

1. Obtain the displacement vector for the element, $\{u_{en}\}$, from the $\{u_{gn}\}$ vectors at its two ends.

2. Evaluate the independent constants $\{q_n\}$ from

$$\{q_n\} = [H_{uq}]^{-1}\{u_{en}\} \quad . \tag{128}$$

3. Evaluate the harmonic components of strain from

$$\left.\begin{aligned} \{\varepsilon_n\} &= [H_{\varepsilon q}]_n\{q_n\} \quad , \\ \{\gamma_n\} &= [H_{\gamma q}]_n\{q_n\} \quad , \\ \{\chi_n\} &= [H_{\chi q}]_n\{q_n\} \quad . \end{aligned}\right\} \tag{129}$$

393

4. Evaluate internal forces from

$$\left.\begin{aligned}\{F_n\} &= t[E]\{\varepsilon_n - \Delta\varepsilon_n\} \quad ,\\ \{V_n\} &= t_s[G]\{\gamma_n\} \quad ,\\ \{M_n\} &= [D]\{\chi_n\} \quad ,\end{aligned}\right\} \tag{130}$$

where $\{\Delta\varepsilon_n\}$ is the incremental strain vector given by Equation 123. $[H_{\varepsilon q}]_n$, $[H_{\chi q}]_n$ and $\{\Delta\varepsilon_n\}$ contain distance along the shell, s, as a parameter. The formulas are to be evaluated for $s = 0$ and $s = \ell$. The procedures for starred components and for $n = 0$ are identical. (No factor of 2 this time).

Membrane stress components are obtained from

$$\{\sigma_n^m\} = [E]\{\varepsilon_n - \Delta\varepsilon_n\} \quad . \tag{131}$$

In evaluating combined membrane and bending stresses at a distance h from the neutral surface in the w direction, it is assumed that the material property is expressed by [E]. Thus

$$\{\sigma_n(h)\} = [E]\{\varepsilon_n - \Delta\varepsilon_n - h\chi_n\} \quad . \tag{132}$$

Components of internal forces and stresses at specified azimuth positions, $\{\phi_i\}$, are obtained by

$$F(\phi_i) = \sum_{n=1}^{m} F_n \cos n\phi_i + \sum_{n=1}^{m} F_n^* \sin n\phi_i + F_0 \quad , \tag{133}$$

for quantities with even symmetry, and by

$$F(\phi_i) = \sum_{n=1}^{m} F_n \sin n\phi_i + F_0^* - \sum_{n=1}^{m} F_n^* \cos n\phi_i \quad , \tag{134}$$

for quantities with odd symmetry.

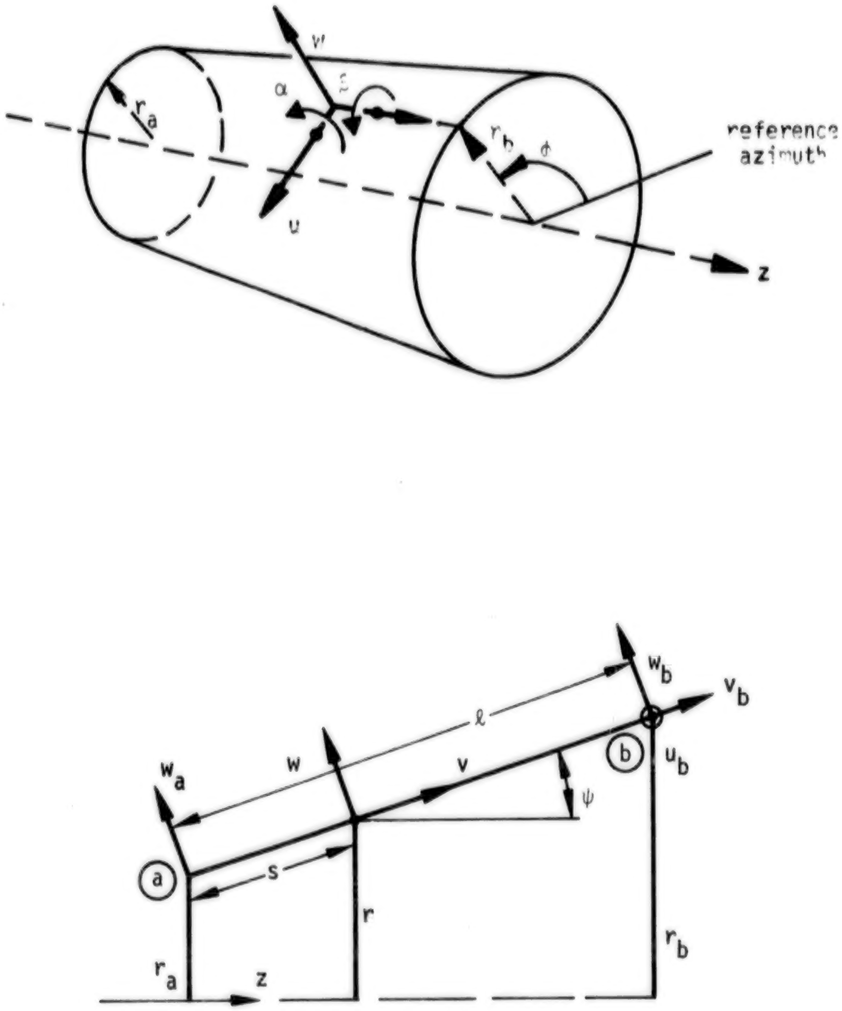
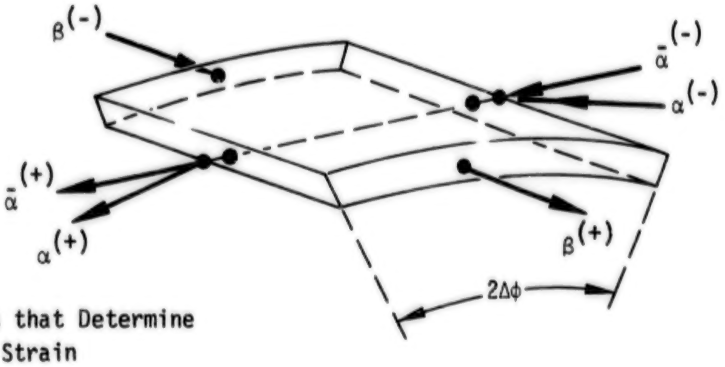
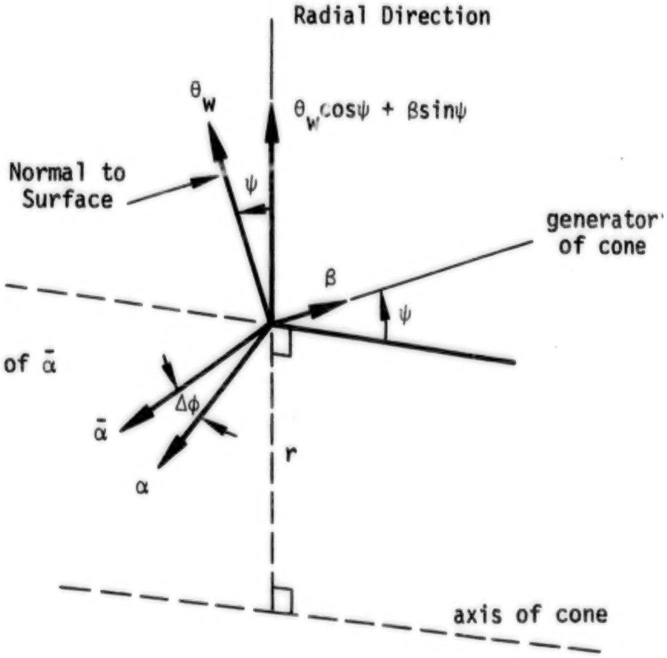

v'
α
β
r_a
r_b
φ
reference azimuth
u
z
w_b
v_b
ℓ
w_a
w
v
ψ
ⓑ
u_b
ⓐ
s
r
r_b
r_a
z

Figure 1. Coordinate geometry for conical shell element.


$\beta^{(-)}$
$\bar{\alpha}^{(-)}$
$\alpha^{(-)}$
$\bar{\alpha}^{(+)}$
$\alpha^{(+)}$
$\beta^{(+)}$
$2\Delta\phi$

(a) Rotations that Determine Twisting Strain


Radial Direction
θ_w
$\theta_w \cos\psi + \beta\sin\psi$
Normal to Surface
ψ
generator of cone
β
ψ
$\bar{\alpha}$
$\Delta\phi$
α
r
axis of cone

(b) Determination of $\bar{\alpha}$

Figure 2. Geometrical relationships used in calculation of twisting strain.

396

5.10 THE DOUBLY CURVED SHELL ELEMENT

5.10.1 Introduction

The formulation of the toroidal ring (and shell cap) elements described herein is derived from, and is mathematically consistent with, the formulation described in Reference 1.

One of the first discrete element models put forward and one which has since been the subject of numerous research investigations and reports is the singly curved ring element formed by a section of revolution of a thin conic shell, (see References 2, 3, 4, 5). The reason for this widespread attention is twofold. First, there exists a broad and important class of axisymmetric thin shell structures which are readily idealized with the conic ring. Second, behavior predictions based on the conic ring have proved, in some cases, to be very poor.

References 6 and 7 have attempted to lay down guidelines in developing shell discrete elements. These references identify the primary sources of difficulty with the associated discontinuities in slope which occur along element circumferential interface lines in the conic shell idealization. Having made this identification, it follows that an advanced ring element is needed which avoids the troublesome discontinuities. The analytical development of the doubly curved shell element used in NASTRAN is presented herein.

A mathematical representation is formulated for a doubly curved ring element. This versatile configuration, defined by an arbitrary section of revolution of a complete right circular toroidal shell, permits a smoothly continuous idealization of general axisymmetric thin shell problems. Section 5.10.3 is devoted to the construction of admissible displacement functions. The importance of selecting appropriate displacement functions cannot be over-emphasized since they serve as a basis for all response characteristics of a discrete element model. Osculatory membrane and hyperosculatory flexure displacement functions which embody generalized degrees of freedom are constructed to achieve smoothness in stress predictions and to minimize the number of discrete elements required in a structural idealization.

A general Lagrangian function is utilized to derive a set of element matrices. The appropriate function is,

$$\frac{\partial \Phi_1}{\partial q_r} + \frac{d}{dt}\left(\frac{\partial \Phi_2}{\partial \dot{q}_r}\right) = 0 \quad , \tag{1}$$

PAGE 397 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

where

q_r = r^{th} generalized displacement coordinate

$\dot{q}_r$ = r^{th} generalized velocity coordinate

Φ_1 = total potential energy

Φ_2 = total kinetic energy

Application of this Lagrange equation yields the following element matrices

1.	Stiffness	$[K]$
2.	Pressure Load	$\{F_P\}$
3.	Thermal Load	$\{F_T\}$
4.	Prestrain Load	$\{F_\varepsilon\}$
5.	Gravity Load	$\{F_G\}$
6.	Stress	$[S]$
7.	Mass	$[M]$

At present other types of structural elements cannot be combined in NASTRAN with the doubly curved shell element in the solution of problems. The loads that are applied to the doubly curved shell element must be axisymmetric. For more general loading conditions the conical shell element described in section 5.9 may be used.

5.10.2 Coordinate System Definition

The basic coordinate system employed is toroidal, which is a right-handed orthogonal curvilinear system as shown in Figure 1. The midplane of the shell is defined by the (α,β) coordinate surface. Principal curvatures of the shell are aligned with the coordinate axes. Complete characterization of the coordinate system is achieved by specification of the metric parameters, A and B, and the principal curvatures of the shell surface, R_α and R_β.

The definition of an increment of length, for a toroidal shell, is

$$(ds)^2 = (d\xi)^2 + (d\eta)^2 \quad , \qquad (1)$$

where $d\xi$ is the increment of length along the meridian, and $d\eta$ is the increment of length along the azimuth. This leads immediately to the definition of the Lamé parameters, i.e., the metric parameters,

$$\begin{aligned} (d\xi)^2 &= (Ad\alpha)^2 \quad , \\ (d\eta)^2 &= (Bd\beta)^2 \quad , \end{aligned} \qquad (2)$$

where α is defined in Figure 1, and β is the rotation about the polar axis.

Restricting consideration to circular cross sections, we first write from observation of Figure 1 the expressions for the coordinate radii of curvature:

$$\begin{aligned} R_\alpha &= a \quad , \\ R_\beta &= a + \frac{b}{\sin\alpha} \quad . \end{aligned} \qquad (3)$$

By virtue of the curvilinear coordinate axes chosen, these are the principal radii of curvature of the shell midsurface. The radius of curvature R_α is simply the radius of the circular cross section. Note that, while R_α is constant, R_β is a function of the α coordinate.

The Lamé parameters of the coordinate system also follow directly from observation of Figure 1:

$$\begin{aligned} A &= a \quad , \\ B &= b + a\sin\alpha \quad . \end{aligned} \qquad (4)$$

400

The conventional characterization of the curvilinear coordinate system, just stated, does not allow convenient transition of the toroidal segment ring element to a conic segment ring element as a special case since (a), and therefore (A), would approach infinity. A specification based on arc length is developed in order to incorporate the conic segment ring element as a special case. With reference to Figure 1, the system characterization is modified as follows:

$$\left.\begin{aligned}
(ds)^2 &= (d\xi)^2 + (Bd\beta)^2 \quad , \\
R_\alpha &= a \quad , \\
R_\beta &= a + \frac{r_1}{\sin(\alpha_1 + \frac{\xi}{a})} - \frac{a \sin \alpha_1}{\sin(\alpha_1 + \frac{\xi}{a})} \quad , \\
A &= 1.0 \quad , \\
B &= r_1 + a \sin(\alpha_1 + \tfrac{\xi}{a}) - a \sin\alpha_1 \quad ,
\end{aligned}\right\} \qquad (5)$$

where

$$0 \le \xi \le a(\alpha_2 - \alpha_1) \quad .$$

An alternate specification is now easily derived for the case of the conic segment ring element. This special form is obtained by rewriting R_β and B as

$$\left.\begin{aligned}
R_\beta &= \frac{r_1}{\sin(\alpha_1 + \frac{\xi}{a})} + \frac{\sin(\alpha_1 + \frac{\xi}{a}) - \sin\alpha_1}{(\frac{1}{a})\sin(\alpha_1 + \frac{\xi}{a})} \quad , \\
B &= r_1 + \frac{\sin(\alpha_1 + \frac{\xi}{a}) - \sin\alpha_1}{(\frac{1}{a})} \quad .
\end{aligned}\right\} \qquad (6)$$

Invoking L'Hospital's rule we obtain the desired characterization of the conic ring

$$R_\alpha = \infty \quad ,$$

$$R_\beta = \frac{r_1}{\sin\alpha_1} + \xi \frac{\cos\alpha_1}{\sin\alpha_1} \quad ,$$

(7)

$$A = 1.0 \quad ,$$

$$B = r_1 + \xi \cos\alpha_1 \quad ,$$

where

$$0 \leq \xi \leq [(r_2 - r_1)^2 + (z_2 - z_1)^2]^{1/2} \quad .$$

Note that further specialization to a cylindrical segment ring is immediate, i.e.,

$$\begin{aligned} R_\alpha &= \infty \quad , \\ R_\beta &= r_1 \quad , \\ A &= 1.0 \quad , \\ B &= r_1 \quad . \end{aligned} \tag{8}$$

Also note that for the special case of the shell cap element, $\alpha_1 = 0$, and $r_1 = 0$, so that in this case

$$\begin{aligned} R_\alpha &= a \quad , \\ R_\beta &= a \quad , \\ A &= 1.0 \quad , \\ B &= a \sin(\tfrac{\xi}{a}) \quad . \end{aligned} \tag{9}$$

5.10.3 Displacement Functions

5.10.3.1 Introduction

The geometric shape and the notation used with the doubly curved ring element are illustrated in Figure 1. By virtue of the assumed axisymmetry of the problem the displacements are functions of a single coordinate. In particular, with reference to Figure 1, the displacements may be written in terms of the meridional arc length coordinate ξ, i.e.

$$\begin{aligned} u(\xi,\beta) &= u(\xi) \quad , \\ w(\xi,\beta) &= w(\xi) \quad . \end{aligned} \tag{1}$$

Explicit consideration is given to the construction of displacement functions which satisfy admissibility requirements. Specifically, admissibility requires that the assumed displacement functions be complete up to the order of truncation, embody all rigid body displacement modes, and provide for interelement continuity. Adherence to this definition of admissibility allows convergence criteria, proven within the framework of continuous mechanics, to be invoked in interpreting predicted behavior.

5.10.3.2 Membrane Displacement Function

The construction of complete displacement functions which provide for interelement continuity is particularly simple in the present problem since the functions are essentially one-dimensional. The proper incorporation of rigid body displacement modes can be verified using the strain displacement relations presented in Section 5.10.4.

Displacement parallel to the meridian will be called "membrane" displacement because it is the principal contributor to membrane strain, see Equation 2 of Section 5.10.4. The membrane displacement function is taken to be a complete cubic in the meridional arc length ξ, i.e.

$$u(\xi) = \beta_1 + \beta_2\xi + \beta_3\xi^2 + \beta_4\xi^3 \quad . \tag{2}$$

The coefficients β_i in the assumed function are referred to as "generalized coordinates". Although only two generalized coordinates are needed to establish the required piecewise continuity of membrane displacement, four are provided by the assumed complete cubic. The two "extra" coefficients are included to obtain improved stress continuity across element interfaces by imposing continuity of the first derivative of displacement.

The use of generalized coordinates as degrees of freedom affords maximum simplification in the development of algebraic expressions for the energy functions; however, in order to apply physical boundary conditions to an element it is necessary to transform to element boundary or grid point degrees of freedom $\{q_{r_m}\}$, i.e.

$$\{\beta_m\} = [\Gamma^{(m)}_{\beta r}]\{q_{r_m}\} \quad , \tag{3}$$

where

$$\{\beta_m\}^T = \lfloor \beta_1, \beta_2, \beta_3, \beta_4 \rfloor \quad , \tag{4}$$

$$\{q_{r_m}\}^T = \lfloor u_1, u_{\xi_1}, u_2, u_{\xi_2} \rfloor \quad , \tag{5}$$

in which the notation implies that

$$u_{\xi_1} = \frac{\partial u}{\partial \xi}\Big|_{\xi=0}, \text{ etc.} \tag{6}$$

The transformation matrix is obtained by imposing the following boundary conditions on the assumed function:

$$u(\xi)|_{\xi=0} = u_1;\ u(\xi)|_{\xi=s} = u_2 \quad , \tag{7}$$

$$u_\xi(\xi)|_{\xi=0} = u_{\xi_1};\ u_\xi(\xi)|_{\xi=s} = u_{\xi_2} \quad . \tag{8}$$

Thus

$$[\Gamma^{(m)}_{\beta r}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{s^2} & -\frac{2}{s} & \frac{3}{s^2} & -\frac{1}{s} \\ \frac{2}{s^3} & \frac{1}{s^2} & -\frac{2}{s^3} & \frac{1}{s^2} \end{bmatrix} \quad . \tag{9}$$

404

It is pertinent to note that, while the above development has been conceptually and algebraically simple, the development of admissible modes for arbitrary two and three dimensional shapes would be much more difficult.

5.10.3.3 Flexural Displacement Function

Displacement normal to the meridian will be called "flexural" displacement because it is the only contributor to flexural curvature, see Equation 3 of Section 5.10.4. The flexural displacement function is taken to be a complete quintic polynomial in the meridional arc length ξ, i.e.

$$w(\xi) = \beta_5 + \beta_6\xi + \beta_7\xi^2 + \beta_8\xi^3 + \beta_9\xi^4 + \beta_{10}\xi^5 \quad . \tag{10}$$

Only four of the six generalized coordinates are required to establish interelement continuity of transverse displacement and slope. As in the case of the membrane displacement function, the additional degrees of freedom are provided to obtain improved stress continuity across element interfaces by imposing continuity of a higher order displacement derivative. Accordingly, continuity of the transverse displacement function value, first derivative, and second derivative is maintained.

The generalized coordinates are employed in the derivation of the element representation. Transformation is then made to grid point degrees of freedom $\{q_{r_f}\}$ i.e.

$$\{\beta_f\} = [\Gamma_{\beta r}^{(f)}]\{q_{r_f}\} \quad , \tag{11}$$

where

$$\{\beta_f\}^T = \lfloor \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10} \rfloor \quad , \tag{12}$$

$$\{q_{r_f}\}^T = \lfloor w_1, w_{\xi_1}, w_{\xi\xi_1}, w_2, w_{\xi_2}, w_{\xi\xi_2} \rfloor \quad . \tag{13}$$

The transformation matrix $[\Gamma_{\beta r}^{(f)}]$, is obtained by imposing the following boundary conditions on the assumed function:

$$w(\xi)|_{\xi=0} = w_1 \quad , \qquad w(\xi)|_{\xi=s} = w_2 \quad ,$$

425

$$w_\xi(\xi)|_{\xi=0} = w_{\xi_1} \ , \quad w_\xi(\xi)|_{\xi=s} = w_{\xi_2} \ ,$$

$$w_{\xi\xi}(\xi)|_{\xi=0} = w_{\xi\xi_1} \ , \quad w_{\xi\xi}(\xi)|_{\xi=s} = w_{\xi\xi_2} \ . \tag{14}$$

Thus

$$[\Gamma_{\beta r}^{(f)}] = \begin{bmatrix} 1 , & 0 , & 0 , & 0 , & 0 , & 0 \\ 0 , & 1 , & 0 , & 0 , & 0 , & 0 \\ 0 , & 0 , & \frac{1}{2} , & 0 , & 0 , & 0 \\ -\frac{10}{s^3} , & -\frac{6}{s^2} , & -\frac{3}{2s} , & \frac{10}{s^3} , & -\frac{4}{s^2} , & \frac{1}{2s} \\ \frac{15}{s^4} , & \frac{8}{s^3} , & \frac{3}{2s^2} , & -\frac{15}{s^4} , & \frac{7}{s^3} , & -\frac{1}{2s^2} \\ -\frac{6}{s^5} , & -\frac{3}{s^4} , & -\frac{1}{2s^3} , & \frac{6}{s^5} , & -\frac{3}{s^4} , & \frac{1}{2s^3} \end{bmatrix} \ . \tag{15}$$

The full transformation from generalized coordinates to displacement degrees of freedom can now be written

$$\{\beta\} = [\Gamma_{\beta r}]\{q\} \ , \tag{16}$$

where

$$\{\beta\}^T = \lfloor \beta_1, \beta_2, \beta_3, \ldots, \beta_{10} \rfloor \ , \tag{17}$$

$$[\Gamma_{\beta r}] = \left[\begin{array}{c:c} \Gamma_{\beta r}^{(m)} & 0 \\ \hdashline 0 & \Gamma_{\beta r}^{(f)} \end{array}\right] \ , \tag{18}$$

$$\{q\}^T = \lfloor u_1, u_{\xi_1}, u_2, u_{\xi_2}, w_1, w_{\xi_1}, w_{\xi\xi_1}, w_2, w_{\xi_2}, w_{\xi\xi_2} \rfloor \ . \tag{19}$$

The reader may recognize the membrane displacement function as a well known Lagrange osculatory interpolation formula and the flexural displacement function as a hyperosculatory interpolation function. The final form of the displacement forms might well have been written immediately without development. The approach taken here was adopted for two reasons. First, it is applicable without conceptual extension to complex elements where standard interpolation formulae are not applicable, and second the generalized coordinates afford considerable algebraic simplification in deriving element representations.

C-5

5.10.4 Potential Energy

Consideration of the potential energy function must be preceded by a statement of the strain-displacement and stress-strain relations appropriate to the axisymmetric thin shell problem. The general strain-displacement relations are:

$$\varepsilon_\alpha = e_\alpha + zH_\alpha \quad ,$$
$$\varepsilon_\beta = e_\beta + zH_\beta \quad , \tag{1}$$

where the membrane strains are

$$e_\alpha = \frac{1}{A}\frac{\partial u}{\partial \alpha} + \frac{1}{R_\alpha} w \quad ,$$
$$e_\beta = \frac{1}{AB}\frac{\partial B}{\partial \alpha} u + \frac{1}{R_\beta} w \quad , \tag{2}$$

and the flexural curvatures are

$$H_\alpha = -\frac{1}{A}\frac{\partial}{\partial \alpha}\left(\frac{1}{A}\frac{\partial w}{\partial \alpha}\right) \quad ,$$
$$H_\beta = \frac{1}{A^2 B}\frac{\partial B}{\partial \alpha}\frac{\partial w}{\partial \alpha} \quad . \tag{3}$$

Introducing the coordinate system characterization derived in Section 5.10.2, the following relationships are obtained:

$$e_\xi = u_\xi + \lambda_1 w \quad ,$$
$$e_\beta = \lambda_2 u + \lambda_3 w \quad , \tag{4}$$

$$H_\xi = - w_{\xi\xi} \quad ,$$
$$H_\beta = - \lambda_2 w \quad , \tag{5}$$

where the notation λ_j is defined as follows:

$$\lambda_1 = \frac{1}{R_\alpha} \quad ,$$

$$\lambda_2 = \frac{1}{B}\frac{\partial B}{\partial \xi} \quad , \tag{6}$$

$$\lambda_3 = \frac{1}{R_\beta} \quad .$$

It must be remembered in giving explicit expression to the λ_j that the degenerate conic and cylindrical configurations require a special form.

Linear elastic material behavior is assumed. In accordance with this assumption a generalized Hooke's law is employed. The elastic material behavior is assumed to take place from an initial state of prestrain, ε_i, and an initial state of prestress, σ_o, i.e.

$$\{\sigma\} - \{\sigma_o\} = [E]\ \{\{\varepsilon\} - \{\varepsilon_i\}\} \quad . \tag{7}$$

The initial stress and strain vectors are assumed to be prescribed. The coefficient matrix [E] is

$$[E] = \frac{1}{1 - \nu_{\alpha\beta}\ \nu_{\beta\alpha}} \begin{bmatrix} E_\alpha & E_\alpha\ \nu_{\beta\alpha} \\ E_\beta\ \nu_{\alpha\beta} & E_\beta \end{bmatrix} \quad . \tag{8}$$

Since the matrix must be symmetric, $E_\alpha\ \nu_{\beta\alpha} = E_\beta\ \nu_{\alpha\beta}$. Note that orthotropic materials are permitted. Having written equations governing strain-displacement and stress-strain, the desired energy function can be constructed. Beginning from the strain energy density definition

$$U' = \int \lfloor d\varepsilon \rfloor \{\sigma\} \quad , \tag{9}$$

the governing stress-strain relation is introduced to obtain

$$U = \int \left[(\tfrac{1}{2})\ \lfloor \varepsilon \rfloor\ [E]\{\varepsilon\} - \lfloor \varepsilon \rfloor\ [E]\{\varepsilon_i\} + \lfloor \varepsilon \rfloor\ \{\sigma_o\}\right) dV \quad , \tag{10}$$

where $\{\varepsilon_i\}$ is the prestrain and $\{\sigma_o\}$ is the prestress.

Carrying forward the previously defined separation of membrane strain {e} and flexural curvature {H} yields, after integration over the shell thickness, the following expression for the potential energy.

$$\begin{aligned} U = \int \Big| & \tfrac{1}{2}\left(\int dz\right) \lfloor e \rfloor\ [E]\{e\} - \left(\int dz\right)\lfloor e \rfloor\ [E]\{e_i\} \\ & + \left(\int dz\right)\lfloor e \rfloor\{\sigma_{mo}\} + \tfrac{1}{2}\left(\int z^2 dz\right) \lfloor H \rfloor\ [E]\{H\} \\ & - \left(\int z^2 dz\right) \lfloor H \rfloor [E]\{H_i\} + \left(\int z^2 dz\right) \lfloor H \rfloor \{\sigma_{fo}\} \Big| dA \quad . \end{aligned} \tag{11}$$

The prestrain and the prestress are related to their membrane and bending components by

$$\begin{aligned} \{\varepsilon_i\} &\equiv \{e_i\} + z\ \{H_i\} \quad , \\ \{\sigma_o\} &\equiv \{\sigma_{mo}\} + z\ \{\sigma_{fo}\} \quad . \end{aligned} \tag{12}$$

Up to this point no assumptions have been made regarding the form of the prestrain and prestress distributions along the meridional coordinate. Explicit consideration will be given to the specification of these quantities subsequently; it is assumed here, however, that the prestrain distribution is linear and that the prestress is constant. The membrane prestrain and the curvature can be expressed in terms of their constant and linear components,

$$\begin{aligned} \{e_i\} &\equiv \{e_i^0\} + \left(\tfrac{\xi}{s}\right)\{e_i{}^1\} \quad , \\ \{H_i\} &\equiv \{H_i^0\} + \left(\tfrac{\xi}{s}\right)\{H_i{}^1\} \quad , \end{aligned} \tag{13}$$

where s is the arc length of the element.

Invoking these assumptions and introducing a convenient symbolic notation, the energy function is rewritten as:

$$\begin{aligned} U = \int_0^s \Big[& \tfrac{1}{2} \lfloor e \rfloor [I_k]\{e\} + \lfloor e \rfloor\ \{I_\sigma\} \\ & - \lfloor e \rfloor \{I_\varepsilon^0\} - \left(\tfrac{\xi}{s}\right) \lfloor e \rfloor \{I_\varepsilon{}^1\} \\ & + \tfrac{1}{2} \lfloor H \rfloor [J_k]\{H\} + \lfloor H \rfloor \{J_\sigma\} \\ & - \lfloor H \rfloor \{J_\varepsilon^0\} - \left(\tfrac{\xi}{s}\right) \lfloor H \rfloor \{J_\varepsilon{}^1\} \Big] B d\xi \quad , \end{aligned} \tag{14}$$

where, with the thickness of the shell represented by t,

$$\begin{aligned}
[I_k] &= 2\pi t[E] \quad , \\
\{I_\sigma\} &= 2\pi t\{\sigma_0\} \quad , \\
\{I_\varepsilon^0\} &= [I_k]\{e_i^0\} \quad , \\
\{I_\varepsilon{}^1\} &= [I_k]\{e_i{}^1\} \quad , \\
[J_k] &= \frac{2\pi t^3}{12}[E] \quad , \\
\{J_\sigma\} &= [J_k]\{\sigma_{f_0}\} \quad , \\
\{J_\varepsilon^0\} &= [J_k]\{H_i^0\} \quad , \\
\{J_\varepsilon{}^1\} &= [J_k]\{H_i{}^1\} \quad .
\end{aligned} \tag{15}$$

5.10.5 Element Stiffness Matrix

The next step in constructing the element representation is to discretize the element by introducing the previously derived displacement functions. The generalized coordinates are transformed to displacement degrees of freedom, and the energy functions are substituted into the Lagrange equation. The element stiffness matrix and the load vectors then arise as stationary conditions of the total potential energy. The stress recovery matrices are derived from equilibrium conditions.

The contribution of linear elastic stiffness to the energy function is, from Equation 14 of Section 5.10.4, omitting terms due to prestress and prestrain

$$\Phi_k = \int_0^s \left[\frac{1}{2} \lfloor e \rfloor [I_k]\{e\} + \frac{1}{2} \lfloor H \rfloor [J_k]\{H\} \right] Bd\xi \quad . \qquad (1)$$

Substituting for strains in terms of displacements, using Equations 4 and 5 of section 5.10.4, yields

$$\begin{aligned} \Phi_k = \int_0^s \Big[& \frac{1}{2} I_{k_{11}} [u_\xi^2 + 2\lambda_1 u_\xi w + \lambda_1^2 w^2] \\ & + \frac{1}{2} I_{k_{12}} [2\lambda_2 u u_\xi + 2\lambda_3 u_\xi w + 2\lambda_1\lambda_2 uw + 2\lambda_1\lambda_3 w^2] \\ & + \frac{1}{2} I_{k_{22}} [\lambda_2^2 u^2 + 2\lambda_2\lambda_3 wu + \lambda_3^2 w^2] \\ & + \frac{1}{2} J_{k_{11}} [w_{\xi\xi}{}^2] \\ & + \frac{1}{2} J_{k_{12}} [2\lambda_2 w_\xi w_{\xi\xi}] \\ & + \frac{1}{2} J_{k_{22}} [\lambda_2^2 w_\xi^2] \Big] Bd\xi \quad . \end{aligned} \qquad (2)$$

The development of an algebraic expression for the elastic strain energy is now pursued by considering each of the energy contributions of Equation 2 individually. That is, each displacement quantity is expanded in terms of the assumed functions, and the indicated integration is carried out.

Addition of all the energy contributions yields

$$\Phi_k = \frac{1}{2} \lfloor \beta \rfloor [\overline{K}] \{\beta\} \quad . \tag{3}$$

Transformation from generalized coordinates to displacement degrees of freedom by means of Equation 16 of Section 5.10.3 produces the element stiffnes matrix

$$[K] = [\Gamma_{\beta r}]^T [\overline{K}] [\Gamma_{\beta r}] \quad . \tag{4}$$

Explicit algebraic representation of the elements of $[\overline{K}]$, which is a 10 x 10 matrix with as many as thirteen terms in each element, is given in Section 8 of the Programmer's Manual. The terms are functions of the following integrals, with the index (j) ranging from zero to ten.

$$\left.\begin{aligned}
\delta_1^j &= \int_0^s \xi^j B d\xi \quad , \\
\delta_2^j &= \int_0^s \xi^j \lambda_2 B d\xi \quad , \\
\delta_3^j &= \int_0^s \xi^j \lambda_3 B d\xi \quad , \\
\delta_4^j &= \int_0^s \xi^j \lambda_2^2 B d\xi \quad , \\
\delta_5^j &= \int_0^s \xi^j \lambda_2 \lambda_3 B d\xi \quad , \\
\delta_6^j &= \int_0^s \xi^j \lambda_3^2 B d\xi \quad .
\end{aligned}\right\} \tag{5}$$

The λ's are defined in Equation 6 of Section 5.10.4. The most general form of B is given by Equation 5 of Section 5.10.2.

5.10.6 Load Vectors

5.10.6.1 Prestrain Load Vector

The prestrain membrane contribution to the potential energy is, from Equation 14 of Section 5.10.4

$$\Phi_{m\varepsilon} = \int_0^s \left[\lfloor e \rfloor \{I_\varepsilon^0\} + \frac{\xi}{s} \lfloor e \rfloor \{I_\xi{}^1\} \right] B d\xi$$

$$= \int_0^s \left[\lfloor e \rfloor [I_k] \{e_i^0\} + \frac{\xi}{s} \lfloor e \rfloor [I]_k \{e_i{}^1\} \right] B d\xi \quad . \tag{1}$$

Substitution of the stress-strain relationships, Equation 4 of Section 5.10.4, and integration yields

$$\Phi_{m\varepsilon} = \lfloor \beta \rfloor [\tilde{F}_{m\varepsilon}^0]^T [I_k] \{e_i^0\} + \lfloor \beta \rfloor [\tilde{F}_{m\varepsilon}{}^1]^T [I_k] \{e_i{}^1\} \quad , \tag{2}$$

where

$$[\tilde{F}_{m\varepsilon}{}^0] = \begin{bmatrix} 0, & \delta_1^0, & 2\delta_1^1, & 3\delta_1^2, & \lambda_1\delta_1^0, & \lambda_1\delta_1^1, & \lambda_1\delta_1^2, & \lambda_1\delta_1^3, & \lambda_1\delta_1^4, & \lambda_1\delta_1^5 \\ \delta_2^0, & \delta_2^1, & \delta_2^2, & \delta_2^3, & \delta_3^0, & \delta_3^1, & \delta_3^2, & \delta_3^3, & \delta_3^4, & \delta_3^5 \end{bmatrix} \quad , \tag{3}$$

$$[\tilde{F}_{m\varepsilon}{}^1] = \frac{1}{s} \begin{bmatrix} 0, & \delta_1^1, & 2\delta_1^2, & 3\delta_1^3, & \lambda_1\delta_1^1, & \lambda_1\delta_1^2, & \lambda_1\delta_1^3, & \lambda_1\delta_1^4, & \lambda_1\delta_1^5, & \lambda_1\delta_1^6 \\ \delta_2^1, & \delta_2^2, & \delta_2^3, & \delta_2^4, & \delta_3^1, & \delta_3^2, & \delta_3^3, & \delta_3^4, & \delta_3^5, & \delta_3^6 \end{bmatrix} \quad . \tag{4}$$

The δ_i^j integrals are defined in Equation 5 of Section 5.10.5.

The prestrain flexure contribution to the potential energy is

$$\Phi_{f\varepsilon} = \int_0^s \left[\lfloor H \rfloor \{J_\varepsilon^0\} + \frac{\xi}{s} \lfloor H \rfloor \{J_\varepsilon{}^1\} \right] B d\xi$$

$$= \int_0^s \left[\lfloor H \rfloor \{J_k\} \{H_i^0\} + \frac{\xi}{s} \lfloor H \rfloor [J_k] \{H_i{}^1\} \right] B d\xi \quad . \tag{5}$$

Substitution of Equation 5 of Section 5.10.4 into Equation 5 yields

$$\Phi_{f\varepsilon} = \lfloor \beta \rfloor \, [\tilde{F}_{f\varepsilon}^{0}]^T [J_k]\{H_i^0\} + \lfloor \beta \rfloor \, [\tilde{F}_{f\varepsilon}{}^{1}]^T [J_k]\{H_i{}^{1}\} \quad , \tag{6}$$

where

$$[\tilde{F}_{f\varepsilon}^{0}] = \begin{bmatrix} 0 \,, 0 \,, 0 \,, 0 \,, 0 \,, & 0, -2\delta_1^0, -6\delta_1^1, -12\delta_1^2, -20\delta_1^3 \\ 0 \,, 0 \,, 0 \,, 0 \,, 0 \,, & -\delta_2^0, -2\delta_2^1, -3\delta_2^2, -\ 4\delta_2^3, -\ 5\delta_2^4 \end{bmatrix} \quad , \tag{7}$$

$$[\tilde{F}_{f\varepsilon}{}^{1}] = \frac{1}{s} \begin{bmatrix} 0 \,, 0 \,, 0 \,, 0 \,, 0 \,, & 0, -2\delta_1^1, -6\delta_1^2, -12\delta_1^3, -20\delta_1^4 \\ 0 \,, 0 \,, 0 \,, 0 \,, 0 \,, & -\delta_2^1, -2\delta_1^2, -3\delta_2^3, -\ 4\delta_2^4, -\ 5\delta_2^5 \end{bmatrix} \quad . \tag{8}$$

Transforming from generalized coordinates to grid point displacement coordinates and substituting into the Lagrange's equation, Equation 1 of Section 5.10.1, the prestrain load vector is obtained.

$$\begin{aligned} \{F_\varepsilon\} &= [\Gamma_{\beta r}]^T [\tilde{F}_{m\varepsilon}^{0}]^T [I_k]\{e_i^0\} + [\Gamma_{\beta r}]^T [\tilde{F}_{m\varepsilon}{}^{1}]^T [I_k]\{e_i{}^{1}\} \\ &+ [\Gamma_{\beta r}]^T [\tilde{F}_{f\varepsilon}^{0}]^T [J_k]\{H_i^0\} + [\Gamma_{\beta r}]^T [\tilde{F}_{f\varepsilon}{}^{1}]^T [J_k]\{H_i{}^{1}\} \quad , \end{aligned} \tag{9}$$

where

$$\{F_\varepsilon\}^T = \lfloor F_{\xi_1}, f_{\xi_1}, F_{\xi_2}, f_{\xi_2}, F_{z_1}, M_{\beta_1}, f_{z_1}, F_{z_2}, M_{\beta_2}, f_{z_2} \rfloor \quad . \tag{10}$$

Note that the f terms correspond to the extra displacement degrees of freedom and have no physical meaning.

5.10.6.2 Pressure Load Vector

The external work done by the normal pressure on the displacements is,

$$W = \int p(\xi)\, w\, (\xi)\, dA \tag{11}$$

Assume a linear pressure distribution with ξ

$$p = p_1 + \frac{p_2 - p_1}{s} \xi \quad . \tag{12}$$

$$dA = 2\pi B d\xi \quad . \tag{13}$$

Thus

$$W = 2\pi \int_0^s p(\xi)\, w\,(\xi)\, B d\xi \quad . \tag{14}$$

Substituting for pressure and displacement yields

$$W = \lfloor \beta \rfloor \{\overline{F}_p\} \quad , \tag{15}$$

where $\{\overline{F}_p\}$ contains only integrals of the type

$$\delta_1^j = \int_0^s \xi^j B d\xi \quad . \tag{16}$$

Explicitly,

$$\{\overline{F}_p\} = 2\pi \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ p_1 \delta_1^0 + \frac{p_2 - p_1}{s} \delta_1^1 \\ p_1 \delta_1^1 + \frac{p_2 - p_1}{s} \delta_1^2 \\ p_1 \delta_1^2 + \frac{p_2 - p_1}{s} \delta_1^3 \\ p_1 \delta_1^3 + \frac{p_2 - p_1}{s} \delta_1^4 \\ p_1 \delta_1^4 + \frac{p_2 - p_1}{s} \delta_1^5 \\ p_1 \delta_1^5 + \frac{p_2 - p_1}{s} \delta_1^6 \end{array} \right\} \quad . \tag{17}$$

Introduction of the transformation from generalized coordinates to grid point displacement degrees of freedom, and substitution into the Lagrange's equation yields the pressure load vector

$$\{F_p\} = [\Gamma_{\beta r}]^T\{\overline{F}_p\} \quad . \tag{18}$$

5.10.6.3 Thermal Load Vector

The thermal load vector is a special case of the prestrain load vector. The necessary change is effected by substitution of the thermal strains into the prestrain vectors.

Four temperatures are given:

$$\begin{aligned} &T_{1i} && \text{inner temperature at grid point 1} \quad , \\ &T_{1o} && \text{outer temperature at grid point 1} \quad , \\ &T_{2i} && \text{inner temperature at grid point 2} \quad , \\ &T_{2o} && \text{outer temperature at grid point 2} \quad . \end{aligned} \tag{19}$$

The thermal membrane strains and the flexural curvatures are

$$\begin{aligned} \{e_T^0\}^T &= \frac{T_{1i} + T_{1o}}{2} \lfloor \alpha_\xi , \alpha_\beta \rfloor \quad , \\ \{e_T^{\,1}\}^T &= \left(\frac{T_{2i} + T_{2o}}{2} - \frac{T_{1i} + T_{1o}}{2}\right) \lfloor \alpha_\xi , \alpha_\beta \rfloor \quad , \\ \{H_T^0\}^T &= \frac{T_{1i} - T_{1o}}{2} \lfloor \alpha_\xi , \alpha_\beta \rfloor \quad , \\ \{H_T^{\,1}\}^T &= \left(\frac{T_{2i} - T_{2o}}{2} - \frac{T_{1i} - T_{1o}}{2}\right) \lfloor \alpha_\xi , \alpha_\beta \rfloor \quad . \end{aligned} \tag{20}$$

Substitution of these expressions into the prestrain load vector, Equation 9, yields the thermal load matrix.

5.10.6.4 Gravity Load Vector

The gravity load vector is obtained by calculating the work done by displacing the element mass. Thus

$$W = \int \rho \, G\overline{w} \, dV \quad . \tag{21}$$

The mass density is given by ρ, and the acceleration of gravity by G. $\overline{w}$ is the deformation along

417

the z axis. This deformation is

$$\bar{w} = w\cos(\alpha_1 + \phi) - u\sin(\alpha_1 + \phi) \quad , \qquad (22)$$

where u and w are given by Equations 2 and 10 of Section 5.10.3.

Substitution into Equation 1 with $dV = tBd\xi d\beta$ yields

$$W = \lfloor \beta \rfloor \{\bar{F}_G\} \quad , \qquad (23)$$

where

$$\{\bar{F}_G\} = 2\pi\rho G t \begin{Bmatrix} -\int_0^s \sin(\alpha_1+\phi)Bd\xi \\ -\int_0^s \sin(\alpha_1+\phi)\xi Bd\xi \\ -\int_0^s \sin(\alpha_1+\phi)\xi^2 Bd\xi \\ -\int_0^s \sin(\alpha_1+\phi)\xi^3 Bd\xi \\ \int_0^s \cos(\alpha_1+\phi)Bd\xi \\ \int_0^s \cos(\alpha_1+\phi)\xi^1 Bd\xi \\ \int_0^s \cos(\alpha_1+\phi)\xi^2 Bd\xi \\ \int_0^s \cos(\alpha_1+\phi)\xi^3 Bd\xi \\ \int_0^s \cos(\alpha_1+\phi)\xi^4 Bd\xi \\ \int_0^s \cos(\alpha_1+\phi)\xi^5 Bd\xi \end{Bmatrix} . \qquad (24)$$

Introduction of the transformation from generalized coordinates to grid point coordinates and substitution into Lagrange's equation yields the gravity load vector

418

$$\{F_G\} = [\Gamma_{\beta r}]^T\{\bar{F}_G\} \quad . \tag{25}$$

5.10.7 Stress Recovery

The element stresses are given by Equation 7 of Section 5.10.4 as

$$\{\sigma\} = [E]\,\{\{\varepsilon\} - \{\varepsilon_t\}\} + \{\sigma_o\} \quad , \tag{1}$$

where

$$\{\sigma\}^T = \lfloor \sigma_\xi, \sigma_\beta \rfloor \quad . \tag{2}$$

For shell problems it is more convenient to work with stress resultants acting in the middle surface of the element. These are given by Novozhilov, Reference (8):

$$\begin{aligned} T_\xi &= \int_{-t/2}^{t/2} \sigma_\xi \left(1 + \frac{z}{R_\beta}\right) dz \quad , \\ T_\beta &= \int_{-t/2}^{t/2} \sigma_\beta \left(1 + \frac{z}{R_\alpha}\right) dz \quad , \\ M_\beta &= \int_{-t/2}^{t/2} \sigma_\xi \left(1 + \frac{z}{R_\beta}\right) z\,dz \quad , \\ M_\xi &= \int_{-t/2}^{t/2} \sigma_\beta \left(1 + \frac{z}{R_\alpha}\right) z\,dz \quad . \end{aligned} \tag{3}$$

In determining the displacements, the contribution of the shear stress $\tau_{\xi z}$ to the strain energy was neglected.

The transverse shear force, Q_ξ, can, however, be evaluated from moment equilibrium, see Figure 2 and Reference (8).

$$Q_\xi = \lambda_2 (M_\beta - M_\xi) + \frac{\partial M_\beta}{\partial \xi} \quad . \tag{4}$$

By substitution of the stress relationships, Equation 1, into Equation 3, and by subsequent substitution into Equation 4, all stress resultants are obtained. Figure 2 shows the positive direction of the stress resultants. The stress resultants arising from elastic stress, prestrain and prestress will be developed separately.

By proper substitution the elastic stress resultants are obtained in terms of the displacements:

$$
\begin{aligned}
T_\xi &= t[E_{11}(u_\xi + \lambda_1 w) + E_{12}(\lambda_2 u + \lambda_3 w)] \quad , \\
T_\beta &= t[E_{12}(u_\xi + \lambda_1 w) + E_{22}(\lambda_2 u + \lambda_3 w)] \quad , \\
M_\beta &= \frac{t^3}{12}[E_{11}(-w_{\xi\xi} + \lambda_1 u_\xi) + E_{12}(-\lambda_2 w_\xi + \lambda_1 \lambda_2 w)] \quad , \\
M_\xi &= \frac{t^3}{12}[E_{12}(-w_{\xi\xi} + \lambda_1 u_\xi) + E_{22}(-\lambda_2 w_\xi + \lambda_1 \lambda_2 w)] \quad , \\
Q_\xi &= \frac{t^3}{12}\{E_{11}\, w_{\xi\xi\xi} - \lambda_2 E_{11}\, w_{\xi\xi} + (\lambda_2^2 E_{22} - \lambda_4 E_{12}) w_\xi\} \quad ,
\end{aligned}
\tag{5}
$$

where

$$\lambda_4 = \frac{1}{B}\frac{\partial^2 B}{\partial \xi^2} \quad , \tag{6}$$

and the simplifying assumption

$$1 + \frac{z}{R_\alpha} \approx 1 + \frac{z}{R_\beta} \approx 1 \quad ,$$

has been made.

Introducing the assumed displacement functions into Equation 5, yields,

$$
\begin{aligned}
\{T_e\} &= [\bar{S}_1]\{\beta\} \quad , \\
\{M_e\} &= [\bar{S}_2]\{\beta\} \quad , \\
Q_e &= \lfloor \bar{S}_3 \rfloor \{\beta\} \quad ,
\end{aligned}
\tag{7}
$$

where

$$\{T_e\}^T = \lfloor T_\xi, T_\beta \rfloor \quad ,$$

$$\{M_e\}^T = \lfloor M_\beta , M_\xi \rfloor \quad . \tag{8}$$

The elements of $[\overline{S}_1]$, $[\overline{S}_2]$ and $[\overline{S}_3]$ are listed in Table 1 on pages 5.10-31 and 5.10-32.

Introduction of the transformation from generalized coordinates to grid point displacements yields the stress resultants

$$\begin{aligned} \{T_e\} &= [S_1]\{q\} \quad , \\ \{M_e\} &= [S_2]\{q\} \quad , \\ Q_e &= \lfloor S_3 \rfloor \{q\} \quad , \end{aligned} \tag{9}$$

where

$$\begin{aligned} [S_1] &= [\overline{S}_1][\Gamma_{\beta r}] \quad , \\ [S_2] &= [\overline{S}_2][\Gamma_{\beta r}] \quad , \\ \lfloor S_3 \rfloor &= \lfloor \overline{S}_3 \rfloor [\Gamma_{\beta r}] \quad . \end{aligned} \tag{10}$$

For the special case of the shell cap element, the following specializations are made

$$\lambda_2 = \frac{\cos(\alpha_1 + \frac{ds}{R_\alpha})}{r_1 - R_\alpha[\sin\alpha_1 - \sin(\alpha_1 + \frac{ds}{R_\alpha})]} \quad , \tag{11}$$

$$\lambda_3 = \frac{1}{R_\alpha} \quad , \tag{12}$$

$$\lambda_4 = -\frac{1}{(R_\alpha)^2} \quad . \tag{13}$$

Equations 4, 5, and 6 are then modified to reflect these changes. In addition, the relationships in Table 1 are replaced by

$$
\begin{aligned}
\tilde{S}_{1,2} &= E_{12} + E_{11} \quad , \\
\tilde{S}_{2,2} &= E_{22} + E_{12} \quad , \\
\tilde{S}_{3,7} &= -2(E_{12} + E_{11}) \quad , \\
\tilde{S}_{4,7} &= 2(E_{22} + E_{12}) \quad , \\
\tilde{S}_{5,8} &= 3(E_{22} - 4E_{11}) \quad .
\end{aligned}
\tag{14}
$$

From Equations 11 and 12 of Section 5.10.4 the prestrain is

$$
\{\varepsilon_i\} = \{\{e_i^0\} + \frac{\xi}{e}\,\{e_i^1\}\} + z\,\{\{H_i\} + \frac{\xi}{s}\,\{H_i^1\}\} \quad , \tag{15}
$$

where

$$
\{\varepsilon_i\}^T = \llcorner \varepsilon_{i\xi},\, \varepsilon_{i\beta} \lrcorner \quad . \tag{16}
$$

The stress induced by prestrain is

$$
\{\sigma_i\} = [E]\{\varepsilon_i\} \quad . \tag{17}
$$

Substitution of Equation 17 into the stress resultants, Equations 3 and 4, produces the prestrain stress resultants

$$
\begin{aligned}
\{T_i\} &= t\{[E]\{e_i^0\} + \frac{\xi}{s}\,[E]\{e_i^1\}\} \quad , \\
\{M_i\} &= \frac{t^3}{12}\,\{[E]\{H_i^0\} + \frac{\xi}{s}\,[E]\{H_i^1\}\} \quad , \\
Q_i &= \frac{t^3}{12}\Big(\lambda_2 \llcorner E_{11} - E_{12},\, E_{12} - E_{22} \lrcorner\ \{\{H_i^0\} + \frac{\xi}{s}\,\{H_i^1\}\} \\
&\qquad + \frac{1}{s}\,\llcorner E_{11},\, E_{12} \lrcorner\ \{H_i^1\}\Big) \quad .
\end{aligned}
\tag{18}
$$

The thermal stress resultants follow immediately by substitution of the thermal strains, Equation 20 of Section 5.10.6, into Equation 18.

The prestress relationship, Equation 12 of Section 5.10.4, is

$$\{\sigma_o\} = \{\sigma_{mo}\} + z\{\sigma_{fo}\} \quad . \tag{19}$$

The prestress stress resultants are given by

$$\begin{aligned} \{T\} &= t\{\sigma_{mo}\} \quad , \\ \{M_o\} &= \frac{t^3}{12}\{\sigma_{fo}\} \quad , \\ Q_o &= 0 \quad . \end{aligned} \tag{20}$$

Finally the complete stress resultants are:

$$\begin{aligned} \{T\} &= \{T_e\} - \{T_i\} + \{T_o\} \quad , \\ \{M\} &= \{M_e\} - \{M_i\} + \{M_o\} \quad , \\ Q &= Q_e - Q_i + Q_o \quad . \end{aligned} \tag{21}$$

Note that the stress resultants at any point within the element may be obtained by substitution of the proper value of the ξ coordinate.

5.10.8 Kinetic Energy and Mass Matrix

The approach taken in expressing the element kinetic energy is the consistent mass approach first put forward by Archer in Reference (9). The same displacement functions employed in representing the element elastic properties are used to find the element kinetic energy. It is further assumed that the rotational energies are small compared to the translational. By virtue of these assumptions, the kinetic energy functional takes the form

$$\Phi_{KE} = 2\pi \int \frac{1}{2} \lfloor \dot{q} \rfloor \lceil \rho \rfloor \{\dot{q}\} B d\xi \quad , \tag{1}$$

where

$$\lceil \rho \rfloor = \lceil \rho_\xi , \rho_z \rfloor$$

are the mass density coefficients in the meridional and normal directions, which need not be equal.

The element mass matrix is obtained by substituting the kinetic energy into Lagrange's equation. Introducing the assumed displacement functions into the kinetic energy functional, Equation 1, yields

$$\Phi_{KE} = \frac{1}{2} \lfloor \dot{\beta} \rfloor [\overline{M}] \{\dot{\beta}\} \quad , \tag{2}$$

where

$$[\overline{M}] = 2\pi t \begin{bmatrix} \overline{M}_{11} & 0 \\ 0 & \overline{M}_{22} \end{bmatrix} \quad , \tag{3}$$

and

$$[\overline{M}_{11}] = \rho_\xi \begin{bmatrix} \delta_1^0 , & \delta_1^1 , & \delta_1^2 , & \delta_1^3 \\ & \delta_1^2 , & \delta_1^3 , & \delta_1^4 \\ & & \delta_1^4 , & \delta_1^5 \\ & & & \delta_1^6 \end{bmatrix} \quad , \tag{4}$$

$$[\overline{M}_{22}] = \rho_z \begin{bmatrix} \delta_1^0, & \delta_1^1, & \delta_1^2, & \delta_1^3, & \delta_1^4, & \delta_1^5 \\ & \delta_1^2, & \delta_1^3, & \delta_1^4, & \delta_1^5, & \delta_1^6 \\ & & \delta_1^4, & \delta_1^5, & \delta_1^6, & \delta_1^7 \\ & & & \delta_1^6, & \delta_1^7, & \delta_1^8 \\ & & & & \delta_1^8, & \delta_1^9 \\ & & & & & \delta_1^{10} \end{bmatrix} . \qquad (5)$$

(See Equation 5 of Section 5.10.5 for definition of δ_1^j.) The transformation from generalized coordinates to displacement degrees of freedom in Lagrange's equation yields the grid point mass matrix.

$$[M] = [\Gamma_{\beta r}]^T[\overline{M}][\Gamma_{\beta r}] \quad . \qquad (6)$$

REFERENCES

1. Mallett, R. H., and Jordan, S., "MAGIC, An Automated General Purpose System for Structural Analysis," Vol. I Engineers Manual, AFFDL TR 68-56 Vol. I, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, January 1969.

2. Meyer, R. R., and Hannon, M. B., "Conical Segment Method for Analyzing Open Crown Shells of Revolution for Edge Loading," AIAA. J., Vol. 1, 1963, pp. 886-891.

3. Grafton, P. E., and Strome, D. R., "Analysis of Axisymmetric Shells by the Direct Stiffness Method," AIAA. J., Vol. 1, No. 10, 1963, pp. 2342-2347.

4. Popov, E. P., Penzien, J., and Lu, Z. A., "Finite Element Solution for Axisymmetric Shells," Proc. ASCE, Jour. of Eng. Mech. Div., October 1964, pp. 119-145.

5. Percy, J. H., Pian, T. H. H., Klein, S., and Navaratna, D. R., "Application of Matrix Displacement Methods to Linear Elastic Analysis of Shells of Revolution," AIAA J., Vol. 3, 1965, pp. 2138-2145.

6. Jones, R. E., and Strome, D. R., "A Survey of the Analysis of Shells by the Displacement Method," Proc. Conf. on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, October 26-28, 1965, AFFDL TR 66-80, 1966.

7. Klein, S., "A Study of the Matrix Displacement Method as Applied to Shells of Revolution," Proc. Conf. on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, October 26-28, 1965, AFFDL TR 66-80, 1966.

8. Novozhilov, V. V., The Theory of Thin Shells, P. Noordhoff Ltd., 1959.

9. Archer, J. S., "Consistent Mass Matrix for Distributed Mass Systems," Proc. ASCE, J. Struct. Div., Vol. 89, 1963, pp. 161-178.

Table 1. Stress Resultant Matrices.

$$[\bar{S}_1]^T = t \left[\begin{array}{l|l}
\lambda_2 E_{12} & \lambda_2 E_{22} \\
\lambda_2 E_{12}\xi + E_{11} & \lambda_2 E_{22}\xi + E_{12} \\
\lambda_2 E_{12}\xi^2 + 2E_{11}\xi & \lambda_2 E_{22}\xi^2 + 2E_{12}\xi \\
\lambda_2 E_{12}\xi^3 + 3E_{11}\xi^2 & \lambda_2 E_{22}\xi^3 + 3E_{12}\xi^2 \\
\lambda_1 E_{11} + \lambda_3 E_{12} & \lambda_1 E_{12} + \lambda_3 E_{22} \\
(\lambda_1 E_{11} + \lambda_3 E_{12})\xi & (\lambda_1 E_{12} + \lambda_3 E_{22})\xi \\
(\lambda_1 E_{11} + \lambda_3 E_{12})\xi^2 & (\lambda_1 E_{12} + \lambda_3 E_{22})\xi^2 \\
(\lambda_1 E_{11} + \lambda_3 E_{12})\xi^3 & (\lambda_1 E_{12} + \lambda_3 E_{22})\xi^3 \\
(\lambda_1 E_{11} + \lambda_3 E_{12})\xi^4 & (\lambda_1 E_{12} + \lambda_3 E_{22})\xi^4 \\
(\lambda_1 E_{11} + \lambda_3 E_{12})\xi^5 & (\lambda_1 E_{12} + \lambda_3 E_{22})\xi^5
\end{array}\right]$$

$$[\bar{S}_2]^T = \frac{t^3}{12} \left[\begin{array}{l|l}
0 & 0 \\
\lambda_1 E_{11} & \lambda_1 E_{12} \\
2\lambda_1 E_{11}\xi & 2\lambda_1 E_{12}\xi \\
3\lambda_1 E_{11}\xi^2 & 3\lambda_1 E_{12}\xi^2 \\
E_{12}\lambda_1\lambda_2 & E_{22}\lambda_1\lambda_2 \\
E_{12}\lambda_1\lambda_2\xi - E_{12}\lambda_2 & E_{22}\lambda_1\lambda_2\xi - E_{22}\lambda_2 \\
E_{12}\lambda_1\lambda_2\xi^2 - 2E_{12}\lambda_2\xi - 2E_{11} & E_{22}\lambda_1\lambda_2\xi^2 - 2E_{22}\lambda_2\xi - 2E_{12} \\
E_{12}\lambda_1\lambda_2\xi^3 - 3E_{12}\lambda_2\xi^2 - 6E_{11}\xi & E_{22}\lambda_1\lambda_2\xi^3 - 3E_{22}\lambda_2\xi^2 - 6E_{12}\xi \\
E_{12}\lambda_1\lambda_2\xi^4 - 4E_{12}\lambda_2\xi^3 - 12E_{11}\xi^2 & E_{22}\lambda_1\lambda_2\xi^4 - 4E_{22}\lambda_2\xi^3 - 12E_{12}\xi^2 \\
E_{12}\lambda_1\lambda_2\xi^5 - 5E_{12}\lambda_2\xi^4 - 20E_{11}\xi^3 & E_{22}\lambda_1\lambda_2\xi^5 - 5E_{22}\lambda_2\xi^4 - 20E_{12}\xi^3
\end{array}\right]$$

Table 1 (Continued)

$$[\bar{S}_3]^T = -\frac{t^3}{12} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\lambda_2^2 E_{22} + \lambda_4 E_{12} \\ 2\xi(-\lambda_2^2 E_{22} + \lambda_4 E_{12}) + E_{11}(2\lambda_2) \\ 3\xi^2(-\lambda_2^2 E_{22} + \lambda_4 E_{12}) + E_{11}(6\xi\lambda_2 + 6) \\ 4\xi^3(-\lambda_2^2 E_{22} + \lambda_4 E_{12}) + E_{11}(12\xi^2\lambda_2 + 24\xi) \\ 5\xi^4(-\lambda_2^2 E_{22} + \lambda_4 E_{12}) + E_{11}(20\xi^3\lambda_2 + 60\xi^2) \end{bmatrix}$$

ORIGINAL PAGE IS OF POOR QUALITY

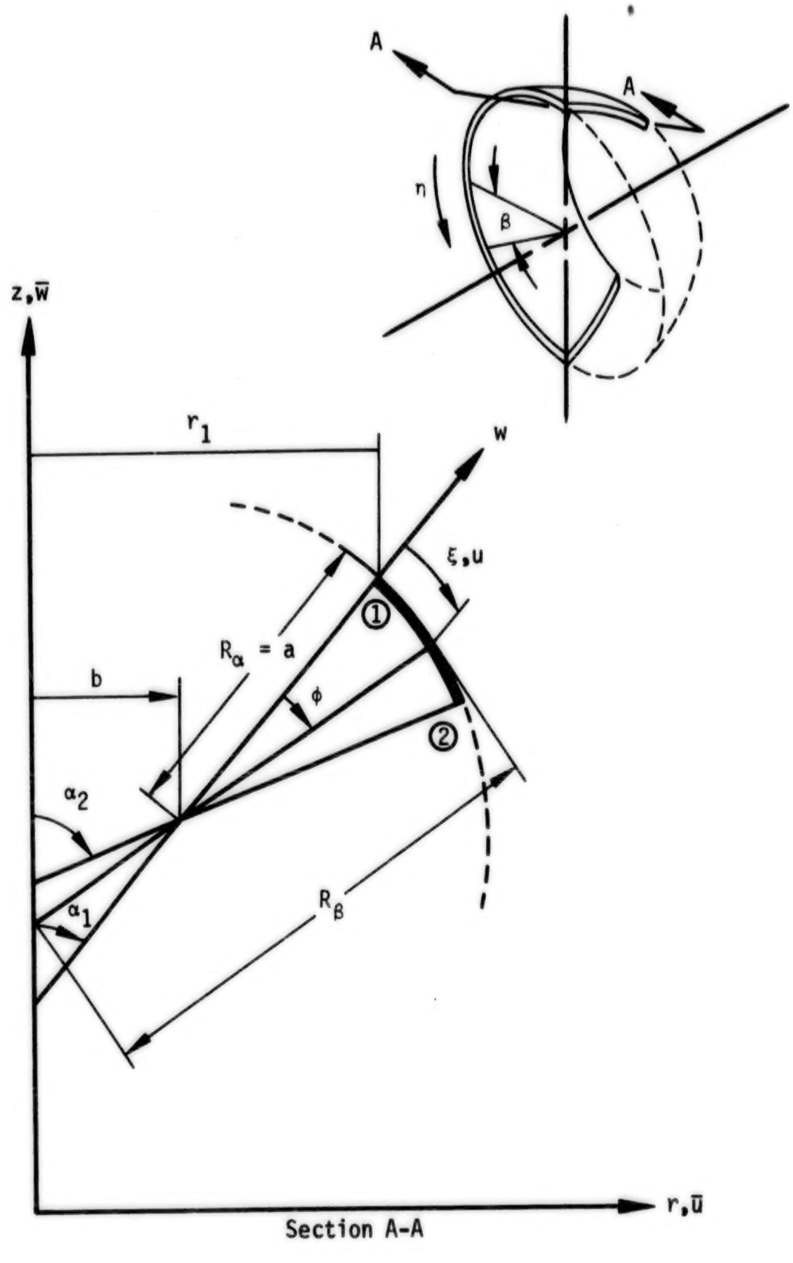

A
A
η
β
z,$\bar{w}$
r_1
w
ξ,u
①
R_α = a
b
φ
②
α_2
α_1
R_β
r,$\bar{u}$
Section A-A

Figure 1. Toroidal thin shell ring representation.

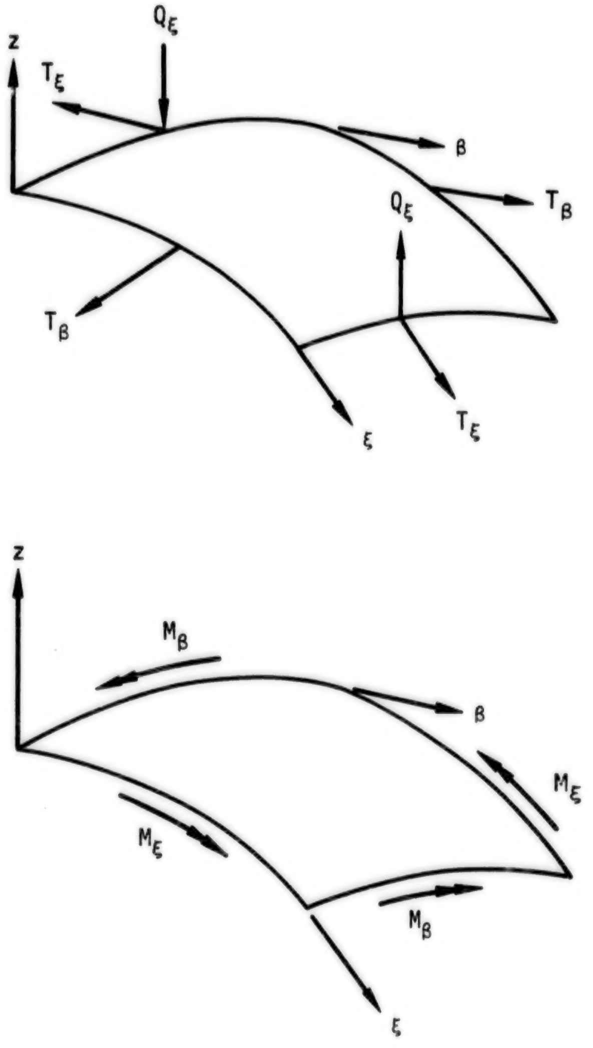

Q_ξ
z
T_ξ
β
T_β
Q_ξ
T_β
ξ
T_ξ
z
M_β
β
M_ξ
M_ξ
M_β
ξ

Figure 2. Definition of stress resultants.

5.11 SOLID OF REVOLUTION ELEMENTS

For the purpose of analyzing axisymmetric structures, NASTRAN includes two different shapes of axisymmetric solid of revolution elements (triangular and trapezoidal) for two different conditions of loading (axisymmetric and nonaxisymmetric). The elements described in this chapter are as follows:

TRIARG - An axisymmetric triangular ring element with axisymmetric loading capability.

TRAPRG - An axisymmetric trapezoidal ring element with axisymmetric loading capability.

TRIAAX - An axisymmetric triangular ring element with nonaxisymmetric loading capability.

TRAPAX - An axisymmetric trapezoidal ring element with nonaxisymmetric loading capability.

The TRIARG and TRAPRG set of elements may be used separately or together and the TRIAAX and TRAPAX set of elements may be used separately or together. These two sets, however, may not be intermixed nor, at present, can either set be combined automatically in NASTRAN with other types of structural elements. In the discussions that follow, the purely axisymmetric elements (TRIARG and TRAPRG) are considered to be degenerate cases of the more general elements capable of nonaxisymmetric loading.

The formulation of these ring elements is derived from, and is mathematically consistent with the formulation described in References 1 and 2. The ring elements provide a powerful tool for the analysis of thick-walled and solid axisymmetric structures of finite length. They may be used to idealize any axisymmetric structure taking into account:

1. arbitrary axial variations in geometry,
2. axial variation in orientation of material axes of orthotropy,
3. radial and axial variations in material properties,
4. any loading system including pressure and temperature, and
5. degradation of material properties due to axisymmetric temperature fields.

The discrete element technique was first applied to the analysis of axisymmetric solids by Clough and Rashid (Reference 3). The formulation of the triangular cross-section ring was extended by Wilson (Reference 4) to include nonaxisymmetric as well as axisymmetric loads. Wilson's formulation is extended here to include orthotropic material properties with variable orientation axes. In addition, the integration of the strain energy over the volume of the ring is effected more precisely. Thermal and pressure load vectors and mass matrices are calculated.

Thus, the discrete element representation presented consists of algebraic expressions for the following matrices;

1. Stiffness , $[K]$
2. Pressure Load , $\{F_p\}$
3. Thermal Load , $\{F_T\}$
4. Gravity Load , $\{F_G\}$
5. Stress , $[S]$
6. Mass , $[M]$

The matrices arise as coefficient matrices in the Lagrange equations for the element. The appropriate generalized form of the Lagrange equation is

$$\frac{\partial \phi_1}{\partial q_r} + \frac{d}{dt}\left(\frac{\partial \phi_2}{\partial \dot{q}_r}\right) = 0 \quad , \tag{1}$$

where

q_r = r^{th} generalized displacement coordinate

ϕ_1 = total potential energy

ϕ_2 = kinetic energy

$\dot{q}_r$ = r^{th} generalized velocity coordinate

The contents of this chapter are organized in the following manner. The description of the basic element geometries and grid point displacement functions are followed by the general development used in formulating the stiffness and mass matrices. Next, the definitions for the material matrix is given and the general equations for the matrices are specialized for the individual elements. The differences that occur are noted in the text. The chapter concludes with the definitions of the load vectors for each element and the stress recovery equations.

5.11.1 General Development for the Axisymmetric Solid Elements

The axisymmetric solid elements provide the NASTRAN user with the means of modeling axisymmetric structures in which three-dimensional, thick-section stress patterns occur. This section describes the general basis for the development of the component matrices. The TRIAAX and TRAPAX elements provide the more general capability by utilizing a Fourier expansion of the load and displacement terms. Their formulation and usage is similar to the conical shell element (CØNEAX) described in Chapter 10. The formulation of the more elementary TRIARG and TRAPRG elements represent the case when only the axisymmetric terms ($n = 0$) are considered. Their formulation differs only in the number of terms to be included in the resulting vectors and matrices. The differences are noted in the text.

5.11.1.1 Coordinate Notation

The coordinate geometry of the axisymmetric ring elements are presented in Figures 1 - 5. The dual coordinate systems illustrated provide for the fact that the element material properties axes may not coincide with the local element axes.

Any point within the element can be located by ascertaining a radial coordinate, r, an axial coordinate, z, and an aximuth coordinate, ϕ. In the TRIAAX and TRAPAX elements the displacement degrees of freedom U_{ri}, $U_{\phi i}$, and U_{zi}, as well as the forces F_{ri}, $F_{\phi i}$, and F_{zi} at each point, i, are assumed to be sinusoidal functions of the circumferential location, ϕ. In the TRIARG and TRAPRG the displacements U_{ri} and U_{zi} are constants with respect to ϕ.

5.11.1.2 Displacement Functions

These ring elements are assumed to be axisymmetric, therefore, the displacements at a position (r,z) can be expanded in a Fourier series with respect to aximuth position, ϕ. The radial, tangential and axial displacements of a point (r,z,ϕ) located within the element can be expressed as

$$\begin{aligned} u(r,z,\phi) &= u_o(r,z) + \sum_{n=1}^{m} u_n(r,z)\cos n\phi + \sum_{n=1}^{m} u_n^*(r,z)\sin n\phi, \\ v(r,z,\phi) &= v_o^*(r,z) + \sum_{n=1}^{m} v_n(r,z)\sin n\phi - \sum_{n=1}^{m} v_n^*(r,z)\cos n\phi, \qquad (2) \\ w(r,z,\phi) &= w_o(r,z) + \sum_{n=1}^{m} w_n(r,z)\cos n\phi + \sum_{n=1}^{m} w_n^*(r,z)\sin n\phi. \end{aligned}$$

The coefficients in the above Fourier series can be interpreted as generalized displacement amplitudes. $u_o(r,z)$ and $w_o(r,z)$ describe axisymmetric motions of the ring, while $v_o^*(r,z)$ describes the displacement in the tangential direction. The displacement field for the TRIARG and TRAPRG elements use only the u_o and w_o displacements and do not use the v_o^* displacements, which describe torsional motion.

Due to the axisymmetry of the ring elements and the resulting orthogonality conditions which arise, the motions corresponding to different harmonic orders (different values of n) are elastically uncoupled. Furthermore, motions represented by starred parameters are not coupled to motions represented by unstarred parameters. For $n > 0$ the stiffness matrices for the starred and unstarred motions are found to be identical. The reason for this is that the starred parameters describe motions that are all shifted $\frac{\pi}{2n}$ in aximuth from the motions described by the unstarred parameters. Thus, $\cos(n\phi) = -\sin(n\phi - \frac{\pi}{2})$ and $\sin(n\phi) = \cos(n\phi - \frac{\pi}{2})$. The unstarred motions will be used to develop the stiffness matrices for $n > 0$. It should be observed that the stiffness matrix for $n = 0$ may include both starred and unstarred motions in the TRIAAX and TRAPAX elements and unstarred motions in the TRIARG and TRAPRG elements.

The degrees of freedom for the ring element are taken to be the values of the Fourier coefficients in Equations 2 evaluated at the nodes of the element. Separate stiffness matrices will be evaluated for the following degrees of freedom:

$$\begin{aligned} \{u_{eo}\}^T &= \lfloor u_{10}, v_{10}^*, w_{10}, u_{20}, v_{20}^*, w_{20} \ \text{- etc. -} \rfloor \\ \{u_{en}\}^T &= \lfloor u_{1n}, v_{1n}, w_{1n}, u_{2n}, v_{2n}, w_{2n} \ \text{- etc. -} \rfloor \end{aligned} \tag{3}$$

where the subscripts 1, 2, and 3 indicate the three nodes of the triangular element (see Figures 1 and 4) or subscripts 1, 2, 3, and 4 indicate the four nodes of the trapezoidal element (see Figures 2 and 5). It should be repeated that for $n > 0$ the stiffness matrices for the starred components will be identical to those of the unstarred components and need not be calculated separately.

5.11.2 Deriving the Stiffness Matrices

The Fourier coefficients in the series presented in Equations 3 can be interpreted as being amplitudes which define the generalized displacements $u(r,z,\phi)$, $v(r,z,\phi)$ and $w(r,z,\phi)$ on the element. These generalized displacement amplitudes can be related to a set of independent constants $\{\beta_n\}$ on the harmonic level, n, via the imposition of an algebraic relationship between the two in the r and z coordinates. Therefore, if $\{\bar{u}_{en}\}^T$ is defined as

$$\{\bar{u}_{en}\}^T = \lfloor u_n(r,z), v_n(r,z), w_n(r,z) \rfloor , \tag{4}$$

then the following relationship may be obtained

$$\{\bar{u}_{en}\} = [H_{\bar{u},\beta}(r,z)]\{\beta_n\} \tag{5}$$

The element strains may be obtained from the displacements by the following standard equations for strain in a cylindrical coordinate system:

$$\begin{aligned}
\varepsilon_{rr} &= \frac{\partial u}{\partial r}\,, \\
\varepsilon_{zz} &= \frac{\partial w}{\partial z}\,, \\
\varepsilon_{\phi\phi} &= \frac{1}{r}\,\frac{\partial v}{\partial \phi} + \frac{u}{r}\,, \\
\varepsilon_{rz} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r}\,, \\
\varepsilon_{r\phi} &= \frac{1}{r}\,\frac{\partial u}{\partial \phi} + \frac{\partial v}{\partial r} - \frac{v}{r}\,, \\
\varepsilon_{z\phi} &= \frac{\partial v}{\partial z} + \frac{1}{r}\,\frac{\partial w}{\partial \phi}\,.
\end{aligned} \tag{6}$$

By substituting Equation 4 into the above functions, and separating the starred quantities, the strain vector for each uncoupled harmonic, in general, is:

$$\{\varepsilon_n\} = \left\{\begin{matrix} \varepsilon_{rr}{}^{n} \\ \varepsilon_{zz}{}^{n} \\ \varepsilon_{\phi\phi}{}^{n} \\ \varepsilon_{rz}{}^{n} \\ \varepsilon_{r\phi}{}^{n} \\ \varepsilon_{z\phi}{}^{n} \end{matrix}\right\} = \left\{\begin{matrix} \dfrac{\partial u_n}{\partial r} \\ \dfrac{\partial w_n}{\partial z} \\ \dfrac{n}{r}\,v_n + \dfrac{u_n}{r} \\ \dfrac{\partial u_n}{\partial z} + \dfrac{\partial w_n}{\partial r} \\ -\dfrac{n}{r}\,u_n + \dfrac{\partial v_n}{\partial r} - \dfrac{v_n}{r} \\ \dfrac{\partial v_n}{\partial z} - \dfrac{n}{r}\,w_n \end{matrix}\right\}. \tag{7}$$

For the TRIARG and TRAPRG elements the strain vector does not contain the last two terms and the v_n terms are omitted. The corresponding starred quantities ($\varepsilon_{rr}{}^{n*}$) have the same form. The strains vary circumferentially according to the following equations:

$$\{\varepsilon(\phi)\} = \{\varepsilon_o\} + \sum_{n=1}^{\infty} [C_n]\,\{\varepsilon_n\} + \sum_{n=1}^{\infty} [C_n{}^*]\{\varepsilon_n{}^*\} \quad , \tag{8}$$

where the diagonal matrices $[C_n]$ and $[C_n{}^*]$, obtained from equations 2 and 6, are

$$\begin{aligned} [C_n] &= [\cos n\phi,\ \cos n\phi,\ \cos n\phi,\ \cos n\phi,\ \sin n\phi,\ \sin n\phi] \ , \\ \text{and} \qquad [C_n{}^*] &= [\sin n\phi,\ \sin n\phi,\ \sin n\phi,\ \sin n\phi,\ -\cos n\phi,\ -\cos n\phi] \ . \end{aligned} \tag{9}$$

In the finite element stiffness method, the displacements are assumed to be a function of the generalized coordinates, α. The strains obtained from Equation 7 in terms of the β coordinates may be expressed by the following matrix equations:

$$\begin{aligned} \{\varepsilon_n\} &= [H_{\varepsilon\alpha}{}^o(r,z)]\{\beta_o\} \quad , \\ \{\varepsilon_n\} &= [H_{\varepsilon\alpha}{}^n(r,z)]\{\beta_n\} \quad , \\ \{\varepsilon_n{}^*\} &= [H_{\varepsilon\alpha}{}^n(r,z)]\{\beta_n{}^*\} \ . \end{aligned} \tag{10}$$

The strain energy, U_e, of the element is expressed as an integral over the volume of the element:

$$U_e = \frac{1}{2} \int_r \int_z \int_o^{2\pi} \{\varepsilon\}^T [E_g] \{\varepsilon\} r\, d\phi\, dz\, dr \ , \tag{11}$$

where the matrix, $[E_g]$, is a basic property of the material defined by Hooke's Law for small displacements, i.e., the basic relationship between the above strain vector and the corresponding stress vector is

$$\{\sigma\} = [E_g]\{\varepsilon\} \ . \tag{12}$$

Substitution of $\{\varepsilon\}$ from Equations 8 and 9 into Equation 11 and integrating with respect to ϕ produces the uncoupling of the strain energy:

$$U_e = U_{eo} + \sum_{n=1}^{m} U_{en} + \sum_{n=1}^{m} U_{en}^{*} , \tag{13}$$

where

$$U_{eo} = \pi \int_r \int_z \{\varepsilon_o\}^T [E_g] \{\varepsilon_o\} \, rdzdr ,$$

$$U_{en} = \frac{\pi}{2} \int_r \int_z \{\varepsilon_n\}^T [E_g] \{\varepsilon_n\} \, rdzdr , \tag{14}$$

and $$U_{en}^{*} = \frac{\pi}{2} \int_r \int_z \{\varepsilon_n^{*}\}^T [E_g] \{\varepsilon_n^{*}\} \, rdzdr .$$

To be consistent with the definition of an axisymmetric material, the last two terms in the strain vector must not be coupled to the other four terms in the vector. When the harmonics can be uncoupled, an inherently three-dimensional problem can be reduced to the consideration of a series of two-dimensional solutions, a phenomenon which occurs due to the axisymmetric character of these elements.

Using Equation 10, Equation 14 can be rewritten as

$$U_{en} = \frac{1}{2} \{\beta_n\}^T [\tilde{K}_n] \{\beta_n\} , \tag{15}$$

where the unstarred harmonic stiffness matrix for the nth harmonic is

$$[\tilde{K}_n] = \pi \int_r \int_z [H_{\varepsilon\alpha}{}^n]^T [E_g] [H_{\varepsilon\alpha}{}^n] \, rdzdr . \tag{16}$$

As was stated in Section 5.11.1, the stiffness matrix for the nth harmonic of the starred series is identical to the nth harmonic, unstarred series and, therefore,

$$[\tilde{K}_n^{*}] = [\tilde{K}_n] , \quad n > 0 . \tag{17}$$

The stiffness matrix for the n = 0 harmonic is multiplied by a factor of two. It consists of uncoupled starred and unstarred partitions. In a symmetric (unstarred analysis) the NASTRAN program limits itself to consideration of only unstarred motions (displacements) and starred motions are eliminated through the use of the single point constraint option. Conversely, in an antisymmetric (starred analysis), all starred motions are eliminated. In the sections which follow, the development of the harmonic stiffness and mass matrices for the solid ring elements is undertaken. Because the stiffness and mass matrices for the n^{th} harmonic starred series are identical to those derived for the n^{th} harmonic unstarred series, distinction between starred and unstarred harmonic quantities are omitted.

5.11.3 Material Definitions

As was previously defined in Equation 12 the $[E_g]$ matrix relates the stresses to the strains and is the means by which the constitutive behavior of the ring element is introduced. Solid materials which exhibit orthogonal anisotropy (orthotropy) in the (r,z) plane can be represented.

Principal directions of elasticity which represent this orthotropy for a given point, are assumed to be in the ϕ, r_m, and z_m directions (see Figure 4). The subscript m indicates a local material axis which is provided to facilitate the description of the properties which characterize a given ring element.

The matrix of elastic constants for an orthotropic body with respect to the (r_m, z_m) coordinate axes for the general TRIAAX and TRAPAX elements is

$$[E_m] = \frac{1}{\Delta}\begin{bmatrix} E_r(1 - \nu_{\phi z}\nu_{z\phi}) & E_r(\nu_{zr} + \nu_{z\phi}\nu_{\phi r}) & E_r(\nu_{\phi r} + \nu_{zr}\nu_{\phi z}) & 0 & 0 & 0 \\ & E_z(1 - \nu_{r\phi}\nu_{\phi r}) & E_\phi(\nu_{z\phi} + \nu_{r\phi}\nu_{zr}) & 0 & 0 & 0 \\ & & E_\phi(1 - \nu_{rz}\nu_{zr}) & 0 & 0 & 0 \\ & & & \Delta G_{rz} & 0 & 0 \\ & \text{(Symmetric)} & & & \Delta G_{r\phi} & 0 \\ & & & & & \Delta G_{z\phi} \end{bmatrix} , \tag{18}$$

where

$$\Delta = 1 - \nu_{r\phi}\nu_{\phi r} - \nu_{\phi z}\nu_{z\phi} - \nu_{zr}\nu_{rz} - \nu_{r\phi}\nu_{\phi z}\nu_{zr} - \nu_{rz}\nu_{\phi r}\nu_{z\phi} , \tag{19}$$

and from symmetry

$$E_r\nu_{\phi r} = E_\phi\nu_{r\phi} \; ; \; E_r\nu_{zr} = E_z\nu_{rz} \; ; \; E_z\nu_{\phi z} = E_\phi\nu_{z\phi} . \tag{20}$$

Poisson's ration, ν_{ij}, is defined as the ratio of the strain in the j directoin to the strain in the i direction due to a stress in the i direction. For the TRIARG and TRAPRG elements, a 4 by 4 matrix, without the last two rows and columns, is used.

If the material axes (r_m, z_m) are oriented at an angle γ from the element geometric axes, see Figure 1 - 5, the following transformation between the harmonic stresses and strains of the two systems must be introduced.

$$\{\epsilon_n^{(m)}\} = [H_\gamma]^T\{\epsilon_n\} \ , \tag{21}$$

and

$$\{\sigma_n\} = [H_\gamma]^T\{\sigma_n^{(m)}\} \ . \tag{22}$$

The transformation matrix [H] is expressed as follows:

$$[H_\gamma] = \begin{bmatrix} \cos^2\gamma & \sin^2\gamma & 0 & -2\sin\gamma\cos\gamma & 0 & 0 \\ \sin^2\gamma & \cos^2\gamma & 0 & 2\sin\gamma\cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \sin\gamma\cos\gamma & -\sin\gamma\cos\gamma & 0 & \cos^2\gamma - \sin^2\gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\gamma & -\sin\gamma \\ 0 & 0 & 0 & 0 & \sin\gamma & \cos\gamma \end{bmatrix} . \tag{23}$$

The transformed material matrix in the cylindrical coordinate system is obtained by the equation

$$[E_g] = [H_\gamma]^T[E_m][H_\gamma] \ . \tag{24}$$

5.11.4 Triangular Ring Elements

Given the general development above, the specific detail of the matrices used for the triangular ring elements can now be presented.

5.11.4.1 Development of the $[H_{\beta,u}]$ Matrix

The generalized displacements $u(r,z,\phi)$, $v(r,z,\phi)$, and $w(r,z,\phi)$ in the radial, tangential, and axial directions of the triangular ring were expressed by Equations 2. The nine displacements for each harmonic as expressed in these equations can be linearly related to a set of nine independent constants. This relationship can be expressed, in general, as

$$
\begin{aligned}
u(r) &= (\beta_1 + \beta_2 r + \beta_3 z) \ , \\
v(r) &= (\beta_4 + \beta_5 r + \beta_6 z) \ , \\
w(r) &= (\beta_7 + \beta_8 r + \beta_9 z) \ .
\end{aligned} \tag{25}
$$

For the TRIARG elements the v motion is zero and only six β terms are used.

Assembling the constants of $\{\beta_n\}$ in vector form, Equations 25 can be rewritten as

$$\{u_e\} = [H_{\beta u}]^{-1}\{\beta\} \ , \tag{26}$$

where the coefficients of $[H_{\beta u}]^{-1}$ are obtained by substituting $u(r,z) = U_i$, $r = r_i$, and $z = z_i$ into Equation 2 for each of the points $i = 1, 2, 3$.

This matrix is then:

$$
[H_{\beta u}]^{-1} = \begin{bmatrix}
1 & r_1 & z_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & r_1 & z_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & r_1 & z_1 \\
1 & r_2 & z_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & r_2 & z_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & r_2 & z_2 \\
1 & r_3 & z_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & r_3 & z_3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & r_3 & z_3
\end{bmatrix} \ . \tag{27}
$$

Columns 2, 5, and 8 and rows 4, 5, and 6 are omitted in the TRIARG formulation. The inversion of Equation 27 provides matrix $[H_{\beta u}]$.

The transformation from the harmonic nodal displacements to the harmonic independent constants to be undertaken can be expressed now as follows:

$$\{\beta_n\} = [H_{\beta u}]\{u_{en}\} \ , \tag{28}$$

5.11.4.2 Development of the $[H_{\varepsilon\beta}]$ Matrix

The strains at any point are functions of the generalized displacements as expressed by Equation 3. The $[H_{\varepsilon\beta}]$ matrix is a function of location and harmonic number and may be obtained from the derivatives of Equation 2 and the definitions of strain for each harmonic, Equation 3. In terms of the generalized coordinate β_i, the strains for each harmonic, n, are

$$\begin{aligned}
\varepsilon_{rr}{}^{n} &= \frac{\partial u}{\partial r} = \beta_2 \,, \\
\varepsilon_{zz}{}^{n} &= \frac{\partial w}{\partial z} = \beta_9 \,, \\
\varepsilon_{\phi\phi}{}^{n} &= \frac{1}{r}\frac{\partial v}{\partial \phi} + \frac{u}{r} = \frac{1}{4}(\beta_1 + n\beta_4) + (\beta_2 + n\beta_5) + \frac{z}{r}(\beta_3 + n\beta_6) \,, \\
\varepsilon_{rz}{}^{n} &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} = \beta_3 + \beta_8 \,, \\
\varepsilon_{r\phi}{}^{n} &= \frac{1}{r}\frac{\partial u}{\partial \phi} + \frac{\partial v}{\partial r} - \frac{v}{r} = -\frac{1}{r}(n\beta_1 + \beta_4) - n\beta_2 - \frac{z}{r}(n\beta_3 + \beta_6) \,,
\end{aligned} \tag{29}$$

and $$\varepsilon_{z\phi}{}^{n} = \frac{\partial v}{\partial z} + \frac{1}{r}\frac{\partial w}{\partial \phi} = -n\frac{\beta_7}{r} + (\beta_6 - n\beta_8) - n\frac{z}{r}\beta_9 \,.$$

Note that the coefficients β_4, β_5, β_6, $\varepsilon_{r\phi}$, and $\varepsilon_{z\phi}$ are uncoupled (torsion) terms, and are the starred coordinate in the zeroth (n = 0) matrix. Thev are not used in the TRIARG element formulation.

In matrix termonology Equations 29 may be expressed as:

$$\{\varepsilon_n\} = [H_{\varepsilon\beta}{}^{n}]\{\beta_n\} \,, \tag{30}$$

where

$$[H_{\varepsilon\beta}{}^{n}] = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\frac{1}{r} & 1 & \frac{z}{r} & \frac{n}{r} & n & \frac{nz}{r} & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
-\frac{n}{r} & -n & -\frac{zn}{r} & -\frac{1}{r} & 0 & -\frac{z}{r} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -\frac{n}{r} & -n & -\frac{nz}{r}
\end{bmatrix} . \tag{31}$$

The middle three columns and the last two rows are not used in the TRAPRG element.

5.11.4.3 Development of the Stiffness Matrix

Equation 16 introduced the expression for the harmonic stiffness matrix $[K_n]$. This result is repeated below:

$$[\tilde{K}_n] = \pi \int_r \int_z [H_{\varepsilon\beta}{}^n]^T [E_g][H_{\varepsilon\beta}{}^n] \, rdzdr \, . \tag{32}$$

The matrices $[H_{\varepsilon\beta}{}^n]$ and $[E_g]$ developed in Section 5.11.3 can now be utilized in the evaluation of $[\tilde{K}_n]$. The triple matrix produce $[H_{\varepsilon\beta}{}^n]^T[E_g][H_{\varepsilon\beta}{}^n]$ which appears in Equation 32 is evaluated in explicit form and integrated by substitution of the integrals which arise in the form

$$\delta_{ij} = \int r^i z^j \, dzdr \, . \tag{33}$$

The harmonic stiffness matrix $[\tilde{K}_n]$ is given in Section 8.25.5 of the NASTRAN Programmer's Manual. The integrals defined in Equation 33 can be written in explicit form and thus numerical integration is not required. Utilizing matrix $[H_{\beta u}]$, the element stiffness matrix in terms of harmonic nodal degrees of freedom $[K_n]$ can now be obtained from the equation

$$[K_n] = [H_{\beta u}]^T[\tilde{K}_n][H_{\beta u}] \, . \tag{34}$$

5.11.4.4 Development of the Mass Matrix

In addition to the stiffness of the triangular ring element, represented by the element stiffness matrix, an essential property in dynamic behavior of the ring is the inertia or mass, represented by the mass matrix. Two basic types of mass matrices will be considered here, these are the lumped mass matrix and the consistent mass matrix.

The lumped mass matrix can be generated in one of two ways:

1. The total mass associated with a nodal circle can be specified directly by an analyst. The program then divides the total mass by 2 for harmonics $n > 0$ and constructs a diagonal (3 x 3) nodal point mass matrix in the usual manner. The total mass is used for the zeroth ($n = 0$) harmonic case, and the TRIARG element.
2. The total mass, specified for a ring, can be equally distributed over the three nodes of the triangular ring element by the program. Each of the diagonal elements on the

representative (3 x 3) grid point mass matrix have the following value for the zeroth harmonic (n = 0):

$$M = \frac{2\pi\rho r_o A}{3} , \tag{35}$$

where

A = Cross-sectional area of the ring element

r_o = Centroidal radius of the ring element

ρ = Density associated with the element as a whole

The diagonal lumped mass matrix described above is multiplied by $(\frac{1}{2})$ for harmonics $n > 0$.

The consistent mass matrix, derived from energy considerations, is developed as follows. The kinetic energy for the triangular ring element can be expressed in the following integral:

$$V = \frac{1}{2}\int_r \rho[\dot{u}(r,z,\phi)^2 + \dot{v}(r,z,\phi)^2 + \dot{w}(r,z,\phi)^2]\, dv \tag{36}$$

where the superscript dot indicates differentiation with respect to time. The harmonic kinetic energy component V_n can be determined following integration of Equation 36 with respect to V.

$$V_j = \frac{\rho\pi}{2}\int_r\int_z \{\dot{\bar{u}}_{en}\}^T\{\dot{\bar{u}}_{en}\}\, rdzdr . \tag{37}$$

Utilizing Equations 5 and 26, Equation 37 can be expressed as follows:

$$V_j = \{\dot{u}_{en}\}^T[H_{\beta u}]^T[\tilde{M}_n][H_{\beta u}]\{\dot{u}_{en}\} , \tag{38}$$

where

$$[\tilde{M}_n] = \frac{\rho\pi}{2}\int_r\int_z [H_{\bar{u}\beta}]^T[H_{\bar{u}\beta}]\, rdzdr , \tag{39}$$

and

$$[H_{\bar{u}\beta}] = \begin{bmatrix} 1 & r & z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & r & z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & r & z \end{bmatrix} \tag{40}$$

The second row and middle three columns correspond to circumferential motion and are omitted in the TRIARG formulation.

The matrix $[\tilde{M}_n]$ can be expressed as given below if the definitions of the integrals defined in Equation 33 are utilized.

$$[\tilde{M}_n] = \frac{\rho\pi}{2} \begin{bmatrix} [\Delta] & & \\ & [\Delta] & \\ & & [\Delta] \end{bmatrix} , \tag{41}$$

where

$$[\Delta] = \begin{bmatrix} \delta_{10} & \delta_{20} & \delta_{11} \\ & \delta_{30} & \delta_{21} \\ & & \delta_{12} \end{bmatrix} . \tag{42}$$

The harmonic consistent mass matrix in terms of nodal circle degrees of freedom for harmonics $n > 0$ can be expressed as

$$[M_n] = [H_{\beta u}]^T [\tilde{M}_n][H_{\beta u}] . \tag{43}$$

For the zeroth harmonic ($n = 0$) matrix $[M_n]$ in Equation 43 should be multiplied by 2. It should be noted that other than this condition, the harmonic consistent mass matrix is not dependent on n.

5.11.5 Trapezoidal Ring Elements

The development procedure utilized in formulating the trapezoidal ring element was previously established in Section 5.11.4 in the formulation of the triangular ring element. This procedure will not be repeated here, but the principal matrices which arise in this second development will be presented.

The generalized harmonic displacements for the trapezoidal ring (analagous to Equation 25 for the triangular ring) are

$$
\begin{aligned}
u_n(r,z) &= \beta_{1n} + \beta_{2n}\, r + \beta_{3n}\, z + \beta_{4n}\, rz \,, \\
v_n(r,z) &= \beta_{5n} + \beta_{6n}\, r + \beta_{7n}\, z + \beta_{8n}\, rz \,, \qquad (44) \\
\text{and} \qquad w_n(r,z) &= \beta_{9n} + \beta_{10n}\, r + \beta_{11n}\, z + \beta_{12n}\, rz \,.
\end{aligned}
$$

In the TRAPRG elements the v motion is zero and only eight β terms are used. Vector $\{u_{en}\}$ for the trapezoidal ring (see Figure 5) appears as

$$\{u_{en}\}^T = [u_{1n}, v_{1n}, w_{1n}, u_{2n}, v_{2n}, v_{2n}, u_{3n}, \ldots, w_{4n}] \,. \qquad (45)$$

The harmonic stiffness matrix for the trapezoidal ring $[K_n]$ can be expressed as

$$[K_n] = [H_{n\beta}]^T [\tilde{K}_n][H_{n\beta}] \,, \qquad (46)$$

where matrix $[\tilde{K}_n]$ is defined by Equation 16.

The harmonic mass matrix for the trapezoidal ring element can be either lumped or consistent in form. Lumped mass considerations are identical to those of the triangular ring element. The consistent mass matrix for the trapezoidal ring can be expressed as

$$[M_n] = [H_{\beta n}]^T [\tilde{M}_n][H_{\beta n}] \,, \qquad (47)$$

where

$$[\tilde{M}_n] = \frac{\rho\pi}{2} \begin{bmatrix} [\Delta] & & \\ & [\Delta] & \\ & & [\Delta] \end{bmatrix} , \qquad (48)$$

and where

$$[\Delta] = \begin{bmatrix} \delta_{10} & \delta_{20} & \delta_{11} & \delta_{21} \\ & \delta_{30} & \delta_{21} & \delta_{31} \\ & & \delta_{12} & \delta_{22} \\ & & & \delta_{32} \end{bmatrix} . \qquad (49)$$

Note that for the TRAPRG elements, the $[\tilde{M}_n]$ matrix contains only eight rows and columns.

5.11.6 Development of Non-Axisymmetric Load Vectors

In the following sections, load vectors resulting from input of temperatures and thermal expansion coefficients are presented for the TRIAAX and TRAPAX ring elements. These loads are presented in a generally 'consistent' framework which implies that the energy or work used to derive the equivalent nodal loads is correlated to the energy (or work) done by the actual applied loads. This concept of "work equivalence" provides a simple framework within which to present the equivalent nodal loads.

Non-axisymmetric direct applied loads, pressure loads, and gravity loads are applied to the TRIAAX and TRAPAX elements with the same method as in the conical shell element (see Section 5.9.9). The equations transforming temperature field definitions to temperature coefficients T_n and T_n^* for each harmonic n, are also given in the same section.

5.11.6.1 The TRIAAX Element Thermal Loads

The thermal load vector for the triangular ring element is constructed assuming uniform distribution of prestrain through the element cross-section. The prestrain contribution to the total potential energy for each harmonic is

$$W_{\varepsilon n} = \pi \int_r \int_z \{\sigma_n\}^T \{\varepsilon_{tn}\} \, rdzdr \ , \tag{50}$$

where

$$\{\varepsilon_{tn}{}^m\} = T_{ne}\{\alpha\} \ . \tag{51}$$

and where $\{\alpha\}$ is the vector of coefficients of thermal expansion and T_{ne} represents the temperature distribution.

$$\{\alpha\}^T = [\alpha_{rr}, \alpha_{zz}, \alpha_{\theta\theta}, 0, 0, 0] \ . \tag{52}$$

Utilizing Equations 5, 10, and 11, Equation 50 can be rewritten as

$$W_{\varepsilon n} = \{u_{\varepsilon n}\}^T [H_{\beta u}]^T \, \pi \int_r \int_z [H_{\varepsilon\beta}]_n^T \, rdzdr \, [E_g]\{\varepsilon_{tn}\} \ . \tag{53}$$

The integral terms in Equation 53 may be expressed as matrix

$$[\bar{H}_{\epsilon\beta}{}^{n}] \equiv \pi \int_{r}\int_{z} [H_{\epsilon\beta}{}^{n}]^{T} \, r\,dz\,dr \; . \tag{54}$$

Since the temperature and material are assumed constant over the cross-section, Equation 52 can be rewritten as

$$W_{\epsilon n} = \{u_{en}\}^{T}[H_{\beta u}]^{T}[H_{\epsilon\beta}]_{n}[E_{g}]\{\epsilon_{tn}\} \; . \tag{55}$$

The harmonic level equivalent nodal loads representing the thermal load vector follow from Equation 55 and can be expressed as

$$\{P_{n}{}^{t}\} = [H_{\beta n}]^{T}[H_{\epsilon\beta}]_{n}[E_{g}]\{\epsilon_{tn}\} \; . \tag{56}$$

Utilizing Equations 51, 52, and 56, the harmonic level equivalent nodal loads representing temperature load can be expressed as

$$\{P_{n}{}^{t}\} = T_{ne}[H_{\beta n}]^{T}[\bar{H}_{\epsilon\beta}]_{n}[E_{g}][H_{\gamma}]^{-1}\{\alpha\} \; , \tag{57}$$

where the order expressed in $\{P_{n}{}^{t}\}$ is established in Equation 3.

The harmonic coefficient, T_{ne}, represents the asymmetrix temperature distribution acting on the element.

5.11.6.2 The TRAPAX Element Thermal Loads

Loading systems applicable to the trapezoidal ring element are generally identical to those developed for the triangular ring element. Differences occur in the expression for the thermal load vector.

The harmonic thermal load vector for the trapezoidal ring can be expressed as

$$\{P_{n}{}^{t}\} = [H_{n\beta}]^{T}[H_{tn}][h]\{T_{n}\} \; , \tag{58}$$

where the matrix $[H_{tn}]$, given in Table 1, is defined as

$$[H_{tn}] = \int_{Vol} [H_{\epsilon\beta}][E_{g}]\{\alpha\}\lfloor 1,r,z,rz \rfloor^{T}$$

and

$$[h] = \frac{1}{A}\begin{bmatrix} r_2z_4(r_3 - r_4) & -r_1z_4(r_3 - r_4) & r_4z_1(r_2 - r_1) & -r_3z_1(r_2 - r_1) \\ -z_4(r_3 - r_4) & z_4(r_3 - r_4) & -z_1(r_2 - r_1) & z_1(r_2 - r_1) \\ -r_2(r_2 - r_4) & r_1(r_3 - r_4) & -r_4(r_2 - r_1) & r_3(r_2 - r_1) \\ (r_3 - r_4) & -(r_3 - r_4) & (r_2 - r_1) & -(r_2 - r_1) \end{bmatrix} , \tag{59}$$

$$A = (r_2 - r_1)(r_3 - r_4)(z_4 - z_1) . \tag{60}$$

The vector T_n represents the harmonic temperatures at each of the four nodes and can be expressed as follows:

$$\{T_n\} = \begin{Bmatrix} T_{1n} - T_o \\ T_{2n} - T_o \\ T_{3n} - T_o \\ T_{4n} - T_o \end{Bmatrix} . \tag{61}$$

and the coefficients are

$$\begin{aligned} A_1 &= E_{11}\alpha_{rr} + E_{12}\alpha_{zz} + E_{13}\alpha_{\theta\theta} , \\ A_2 &= E_{12}\alpha_{rr} + E_{22}\alpha_{zz} + E_{23}\alpha_{\theta\theta} , \\ A_3 &= E_{13}\alpha_{rr} + E_{23}\alpha_{zz} + E_{33}\alpha_{\theta\theta} , \\ \text{and} \quad A_4 &= E_{14}\alpha_{rr} + E_{24}\alpha_{zz} + E_{34}\alpha_{\theta\theta} . \end{aligned} \tag{62}$$

5.11.7 Load Vectors for the TRIARG and TRAPRG Elements

5.11.7.1 Pressure Vectors for the TRIARG Elements

The following indicates the procedure for deriving forces corresponding to a pressure loading. However, this procedure has not been implemented in the program.

Assume a linearly varying normal pressure distribution on the boundary between corners in Figure 1. Note that the direction of the load is determined by the order of the grid points so that the grid points should be numbered consistently. Counterclockwise order is assumed by the program. For side ① , ② ,

$$p(r,z) = p_1 + a_1r + a_2z , \tag{63}$$

Table 1. Matrix $[H_{tn}]$.

$$
\begin{bmatrix}
A_3\,\delta_{0,0} & A_3\,\delta_{1,0} & A_3\,\delta_{0,1} & A_3\,\delta_{1,1} \\
(A_1 + A_3)\delta_{1,0} & (A_1 + A_3)\delta_{2,0} & (A_1 + A_3)\delta_{1,1} & (A_1 + A_3)\delta_{2,1} \\
A_3\,\delta_{0,1} + A_4\,\delta_{1,0} & A_3\,\delta_{1,1} + A_4\,\delta_{2,0} & A_3\,\delta_{0,2} + A_4\,\delta_{1,1} & A_3\,\delta_{1,2} + A_4\,\delta_{2,1} \\
(A_1 + A_3)\delta_{1,1} + A_4\,\delta_{2,0} & (A_1 + A_3)\delta_{2,1} + A_4\,\delta_{3,0} & (A_1 + A_3)\delta_{1,2} + A_4\,\delta_{2,1} & (A_1 + A_3)\delta_{2,2} + A_4\,\delta_{3,1} \\
jA_3\,\delta_{0,0} & jA_3\,\delta_{1,0} & jA_3\,\delta_{0,1} & jA_3\,\delta_{1,1} \\
jA_3\,\delta_{1,0} & jA_3\,\delta_{2,0} & jA_3\,\delta_{1,1} & jA_3\,\delta_{2,1} \\
jA_3\,\delta_{0,1} & jA_3\,\delta_{1,1} & jA_3\,\delta_{0,2} & jA_3\,\delta_{1,2} \\
jA_3\,\delta_{1,1} & jA_3\,\delta_{2,1} & jA_3\,\delta_{1,2} & jA_3\,\delta_{2,2} \\
0 & 0 & 0 & 0 \\
A_4\delta_{1,0} & A_4\delta_{2,0} & A_4\delta_{1,1} & A_4\delta_{2,1} \\
A_2\delta_{1,0} & A_2\delta_{2,0} & A_2\delta_{1,1} & A_2\delta_{2,1} \\
A_2\delta_{2,0} + A_4\delta_{1,1} & A_2\delta_{3,0} + A_4\delta_{2,1} & A_2\delta_{2,1} + A_4\delta_{1,2} & A_2\delta_{3,1} + A_4\delta_{2,2}
\end{bmatrix}
$$

where

$$a_1 = -\frac{(p_2 - p_1)z_1}{r_1 z_2 - r_2 z_1} , \tag{64}$$

$$a_2 = \frac{(p_2 - p_1)r_1}{r_1 z_2 - r_2 z_1} . \tag{65}$$

The external work done by the pressure is

$$W = \int (p_r u + p_z w)dA ,$$

where

$$\begin{aligned} p_r &= -p \sin\alpha \\ p_z &= p \cos\alpha \\ dA &= 2\pi r ds \\ ds &= \sqrt{dr^2 + dz^2} = \frac{dr}{\cos\alpha} . \end{aligned} \tag{66}$$

Substituting back into the work equation

$$W = \int_{r_1}^{r_2} [-(p_1+a_1 r+a_2 z)(\beta_1+\beta_2 r+\beta_3 z)\sin\alpha+(p_1+a_1 r+a_2 z)(\beta_4+\beta_5+\beta_6 z)\cos\alpha]2\ r\frac{dr}{\cos\alpha} . \tag{67}$$

This expression results in three definite integrals

$$\delta_1 = \int_{r_1}^{r_2} rdr = \frac{r_2{}^2-r_1{}^2}{2};\ \delta_2 = \int_{r_1}^{r_2} r^2 dr = \frac{r_2{}^3-r_1{}^3}{3};\ \delta_3 = \int_{r_1}^{r_2} r^3 dr = \frac{r_2{}^4-r_1{}^4}{4} . \tag{68}$$

Substituting into Equation 67 we may write the work equation as

$$W = [\beta]\{\bar{F}_p\} , \tag{69}$$

where $\{\bar{F}_p\}$, the forces on the generalized coordinates, are

ORIGINAL PAGE IS
OF POOR QUALITY

$$\{\bar{F}_p\} = 2\pi \begin{Bmatrix} -k_{12}[(p_1+a_2m_{12})\delta_1+(a_1+a_2k_{12})\delta_2] \\ -k_{12}[(p_1+a_2m_{12})\delta_2+(a_1-a_2k_{12})\delta_3] \\ -k_{12}[(p_1+a_2m_{12})m_{12}\delta_1+\{p_1k_{12}+m_{12}(a_12a_2k_{12})\}\delta_2+(a_1+a_2k_{12})k_{12}\delta_3] \\ (p_1+a_2m_{12})\delta_1+(a_1+a_2k_{12})\delta_2 \\ (p_1+a_2m_{12})\delta_2+(a_1+a_2k_{12})\delta_3 \\ (p_1+a_2m_{12})m_{12}\delta_1+\{p_1k_{12}+m_{12}(a_1+2a_2k_{12})\}\delta_2+(a_1+a_2k_{12})k_{12}\delta_3 \; . \end{Bmatrix} \qquad (70)$$

A special case is obtained when $r_2 = r_1$ and the formulation must be changed. Equation 65 becomes

$$W = \int p_r u dA \; , \qquad (71)$$

where

$$p_r = -(p_1+a_1r_1+a_2z)\sin\alpha \quad , \qquad (72)$$

thus,

$$W = \int_{z_1}^{z_2} -(p_1+a_1r_1+a_2z)(\beta_1+\beta_2r_1+\beta_3z)2\pi r_1 dz \quad . \qquad (73)$$

Note that α may take two values, 90° and 270°, and sin α changes sign accordingly. There will be three definite integrals:

$$\delta_4 = \int_{z_1}^{z_2} dz; \; \delta_5 = \int_{z_1}^{z_2} zdz; \; \delta_6 = \int_{z_1}^{z_2} z^2 dz \quad . \qquad (74)$$

The generalized force vector is

$$\{\bar{F}_p\} = 2\pi r_1 \begin{Bmatrix} -[(p_1+a_1r_1)\delta_4+a_2\delta_5] \\ -[(p_1+a_1r_1)r_1\delta_4+a_2r_1\delta_5] \\ -[(p_1+a_1r_1)\delta_5+a_2\delta_6] \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{75}$$

Transforming from generalized coordinates to grid point displacement coordinates and substituting into the Lagrange equation, we obtain the pressure load vector

$$\{F_p\} = [H_{\beta u}]^T\{\bar{F}_p\} \quad , \tag{76}$$

where

$$F_p^{\ T} = \left\lfloor F_{p_{r_1}}, F_{p_{z_1}}, F_{p_{r_2}}, F_{p_{z_2}}, F_{p_{r_3}}, F_{p_{z_3}} \right\rfloor . \tag{77}$$

5.11.7.2 Prestrain and Thermal Load Vectors

The prestrain load vector is constructed assuming uniform distribution of prestrain across the element. The prestrain contribution to the total potential energy is

$$\Phi_\varepsilon = \int \{\varepsilon\}^T[E]\{\varepsilon_i\}dV \quad . \tag{78}$$

Substitute Equations 26 and 29 of Section 5.11.4 into Equation 78,

$$\Phi_\varepsilon = \{\beta\}^T\pi\int [D]^T rdzdr[E]\{\varepsilon_i\} \quad . \tag{79}$$

Let

$$[\bar{D}] = \pi\int[D]rdzdr \quad , \tag{80}$$

which may be written in terms of the δ_{ij} integrals, as

ORIGINAL PAGE IS OF POOR QUALITY

$$[\bar{D}] = \pi \begin{bmatrix} 0 , & \delta_{10}, & 0 , & 0 , & 0 , & 0 \\ \delta_{00}, & \delta_{10}, & \delta_{01}, & 0 , & 0 , & 0 \\ 0 , & 0 , & 0 , & 0 , & 0 , & \delta_{10} \\ 0 , & 0 , & \delta_{10}, & 0 , & \delta_{10}, & 0 \end{bmatrix} . \qquad (81)$$

For harmonic zero the result is multiplied by two.

Transformation of Equation 79 to grid point displacement coordinates and substitution into the Lagrange equation yields the prestrain load vector

$$\{F_\varepsilon\} = [H_{\beta u}]^T[D]^T[E]\{\varepsilon_i\} \ , \qquad (82)$$

where the load components are

$$\{F_\varepsilon\}^T = \left\lfloor F_{\varepsilon_{r_1}}, F_{\varepsilon_{z_1}}, F_{\varepsilon_{r_2}}, F_{\varepsilon_{z_2}}, F_{\varepsilon_{r_3}} \right\rfloor , \qquad (83)$$

and the prestrain components are

$$\{\varepsilon_i\}^T = \left\lfloor \varepsilon_{i_r}, \varepsilon_{i_\theta}, \varepsilon_{i_z}, 0 \right\rfloor . \qquad (84)$$

The thermal load vector is a special case of the prestrain load vector. Let the initial or thermal strain be

$$\{\bar{\alpha}\}^T = \Delta T \left\lfloor \alpha_r, \alpha_\theta, \alpha_z, 0 \right\rfloor \ , \qquad (85)$$

where ΔT is the average temperature rise of the element above ambient, obtained by averaging the adjacent grid point temperatures.

Substitution into Equation 82 yields the thermal load vector.

$$\{F_T\} = [H_{\beta q}]^T[D]^T[E]\{\bar{\alpha}\} \ . \qquad (86)$$

5.11.7.3 Load Vectors for the TRAPRG Element

The load vectors for the TRAPRG elements are evaluated in the same general manner as for the triangular ring element.

The pressure load is assumed to vary linearly along each edge, as shown in Figure 2. The equivalent loads at grid points are derived from the formula for the work done by pressure

$$W = \int (p_r u + p_z w) dA \quad . \tag{87}$$

u and w are replaced by their expansions in terms of the generalized coordinates, Equations 44 of Section 5.11.5. The coefficients of the β's in the resulting equation are the forces due to pressure on the generalized coordinates, $\{\bar{F}_p\}$. The loads on grid points are obtained by the transformation

$$\{F_p\} = [H_{\beta u}]^T \{\bar{F}_p\} \quad . \tag{88}$$

The prestrain load vector is computed in the same manner as for the triangular ring element, Equation 82. The thermal strain vector is treated differently than in the case of the triangular ring element in that temperature is permitted to be a function of position. Specifically,

$$T(r,z) = K_1 + K_2 r + K_3 z + K_4 rz \quad , \tag{89}$$

where the K's are evaluated so that Equation 89 gives the known temperatures at the grid points.

The contribution of thermal prestrain to the total potential energy is

$$\phi_\varepsilon = \int \{\varepsilon\}^T [E] \{\varepsilon_i\} dV \quad , \tag{90}$$

where

$$\{\varepsilon_i\} = T \lfloor \alpha_r, \alpha_\theta, \alpha_z, 0 \rfloor^T = T\{\bar{\alpha}\} \quad , \tag{91}$$

in which the α's are thermal expansion coefficients. Upon substitution for $\{\varepsilon_i\}$ from Equation 91 and replacement of $\{\varepsilon\}$ by its relationship to grid point displacements, Equation 90 becomes

$$\begin{aligned} \phi_\varepsilon &= \{q\}^T [H_{\beta q}]^T \left[\int [D]^T T dV \right] [E] \{\bar{\alpha}\} \quad , \\ &= \{q\}^T \{F_T\} \quad , \end{aligned} \tag{92}$$

where $\{F_T\}$ is the equivalent thermal load vector at grid points. The integral in Equation 92 is easily evaluated in terms of the δ_{ij} coefficients.

5.11.8 Stress and Displacement Recovery

Displacement information for the triangular and trapezoidal ring elements can be requested in two forms:

1. Harmonic coefficients of motion (displacement amplitudes) as expressed in Equation 2, the TRIARG and TRAPRG elements produce only the zero harmonic (n = 0).
2. Displacements at specified locations, ϕ, on a nodal circle obtained by evaluating Equation 2 for each n for the TRIAAX and TRAPAX elements only.

Stresses for the ring elements are calculated from the expressions

$$\begin{aligned}
\sigma_{rr} &= \sigma_{rr}{}^{0} + \sum_{n=1} (\sigma_{rr}{}^{n} \cos n\phi + \sigma_{rr}{}^{n*} \sin n\phi) \quad , \\
\sigma_{zz} &= \sigma_{zz}{}^{0} + \sum_{n=1} (\sigma_{zz}{}^{n} \cos n\phi + \sigma_{zz}{}^{n*} \sin n\phi) \quad , \\
\sigma_{\phi\phi} &= \sigma_{\phi\phi}{}^{0} + \sum_{n=1} (\sigma_{\phi\phi}{}^{n} \cos n\phi + \sigma_{\phi\phi}{}^{n*} \sin n\phi) \quad , \\
\tau_{rz} &= \tau_{rz}{}^{0} + \sum_{n=1} (\tau_{rz}{}^{n} \cos n\phi + \tau_{rz}{}^{n*} \sin n\phi) \quad , \\
\tau_{r\phi} &= \tau_{r\phi}{}^{0*} + \sum_{n=1} (\tau_{r\phi}{}^{n} \sin n\phi - \tau_{r\phi}{}^{n*} \cos n\phi) \quad , \\
\tau_{z\phi} &= \tau_{z\phi}{}^{0*} + \sum_{n=1} (\tau_{z\phi}{}^{n} \sin n\phi - \tau_{z\phi}{}^{n*} \cos n\phi) \quad .
\end{aligned} \tag{93}$$

where for the n^{th} harmonic, unstarred series

$$\{\sigma_n\} = [E_g]\Big[[H_{\varepsilon\beta}]_n [H_{\beta u}]\{u_{en}\} - \{\varepsilon_{tn}{}^{m}\} \Big] \quad . \tag{94}$$

The matrices and vectors utilized in Equation 94 above are provided by Equations 5, 10, 12, and 51. Equation 94 is evaluated at the centroid of the triangular ring element and stresses can be provided as

1. Harmonic coefficients (stress amplitudes) as expressed in Equation 94.
2. Stresses at specified circumferential locations ϕ, on a circle generated by the element centroid, are obtained by evaluating Equation 93 for each ϕ.

Stresses for the trapezoidal ring element can be recovered by a similar procedure with the exception that for the trapezoidal ring element, stresses are evaluated at the four nodal regions as well as at a fifth region which corresponds to a point that is obtained by averaging the coordinates of the four nodal points.

REFERENCES

1. Mallett, R. H. and Jordan, S., "MAGIC, An Automated General Purpose System for Structural Analysis," Vol. I, Engineers Manual, AFFDL-TR-68-56, Vol. I, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, January 1969.

2. Jordan, S., "MAGIC II, An Automated General Purpose System for Structural Analysis," Vol. I, Engineers Manual (Addendum) AFFDL-TR-71-1, Vol. I, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, May 1971.

3. Clough, R. W. and Rashid, Y., "Finite Element Analysis of Axisymmetric Solids," Proc. ASCE, Journal of Engineering Mechanics Division, Vol. 91, 1965, pp. 71-85.

4. Wilson, E. L., "Structural Analysis of Axisymmetric Solids," AIAA Journal, Vol. 3, No. 12, December 1965, pp. 2269-2274.

ORIGINAL PAGE IS
OF POOR QUALITY

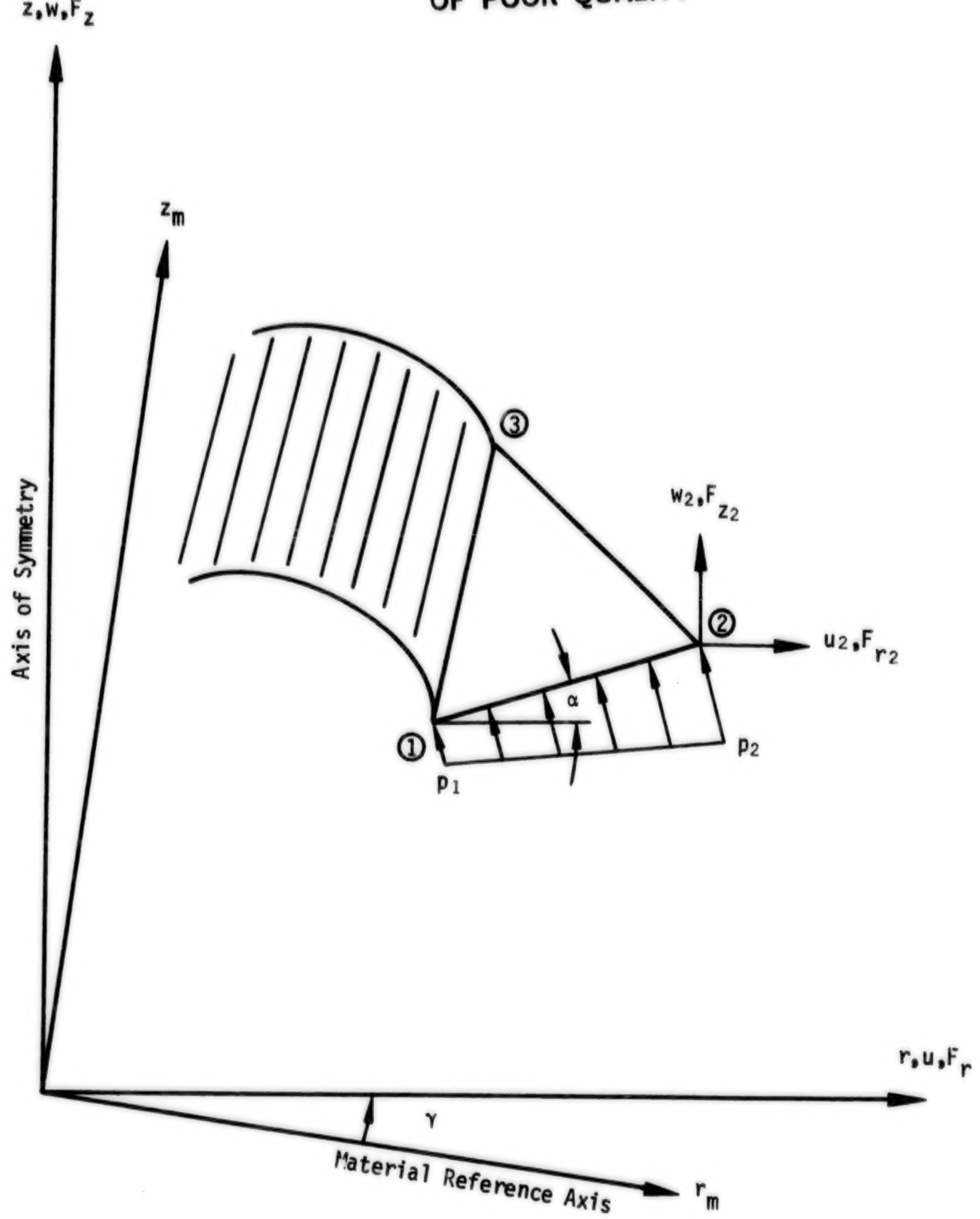

z,w,F$_z$
z_m
Axis of Symmetry
③
w_2,F_{z2}
②
u_2,F_{r2}
α
①
p_1
p_2
r,u,F$_r$
γ
Material Reference Axis
r_m

Figure 1. Triangular ring element.

ORIGINAL PAGE IS
OF POOR QUALITY

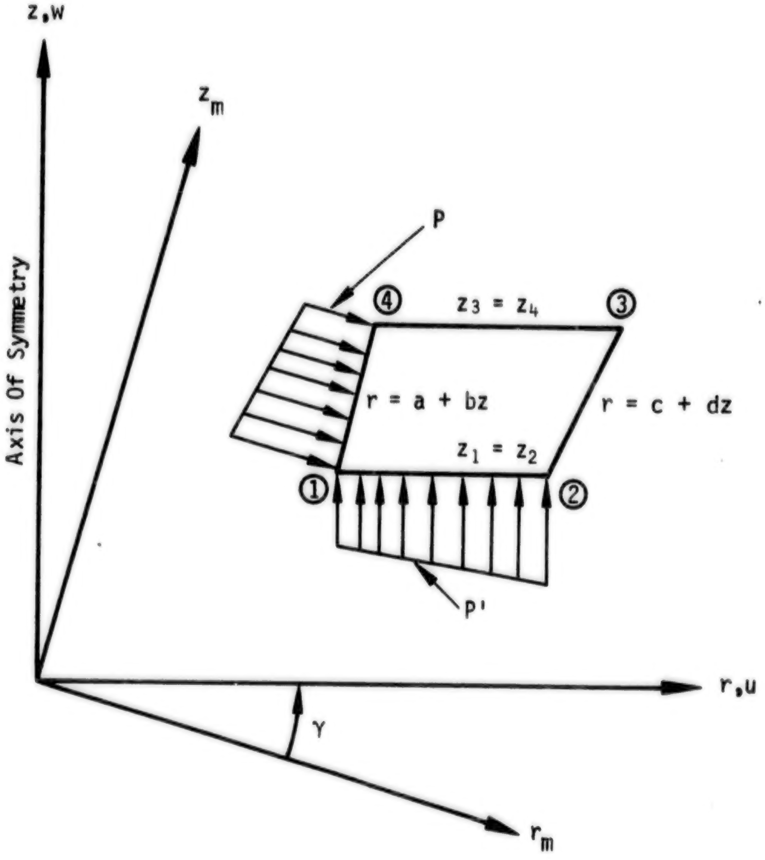
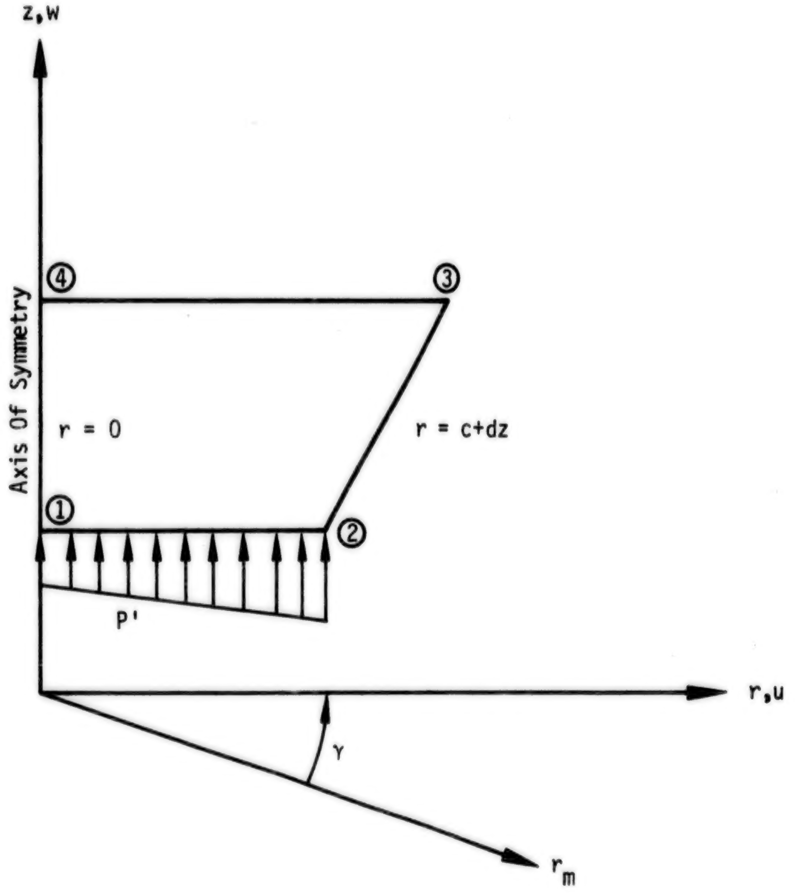

z,w
z_m
Axis Of Symmetry
P
④
$z_3 = z_4$
③
$r = a + bz$
$r = c + dz$
$z_1 = z_2$
①
②
P'
r,u
γ
r_m

Figure 2. Trapezoidal ring element.


z,w
Axis Of Symmetry
④
③
r = 0
r = c+dz
①
②
P'
r,u
γ
r_m

Special Conditions On Core Element

(a) $r_1 = r_4 = 0$

(b) $u_1 = u_4 = 0$

Figure 3. Core element specialization of TRAPRG.

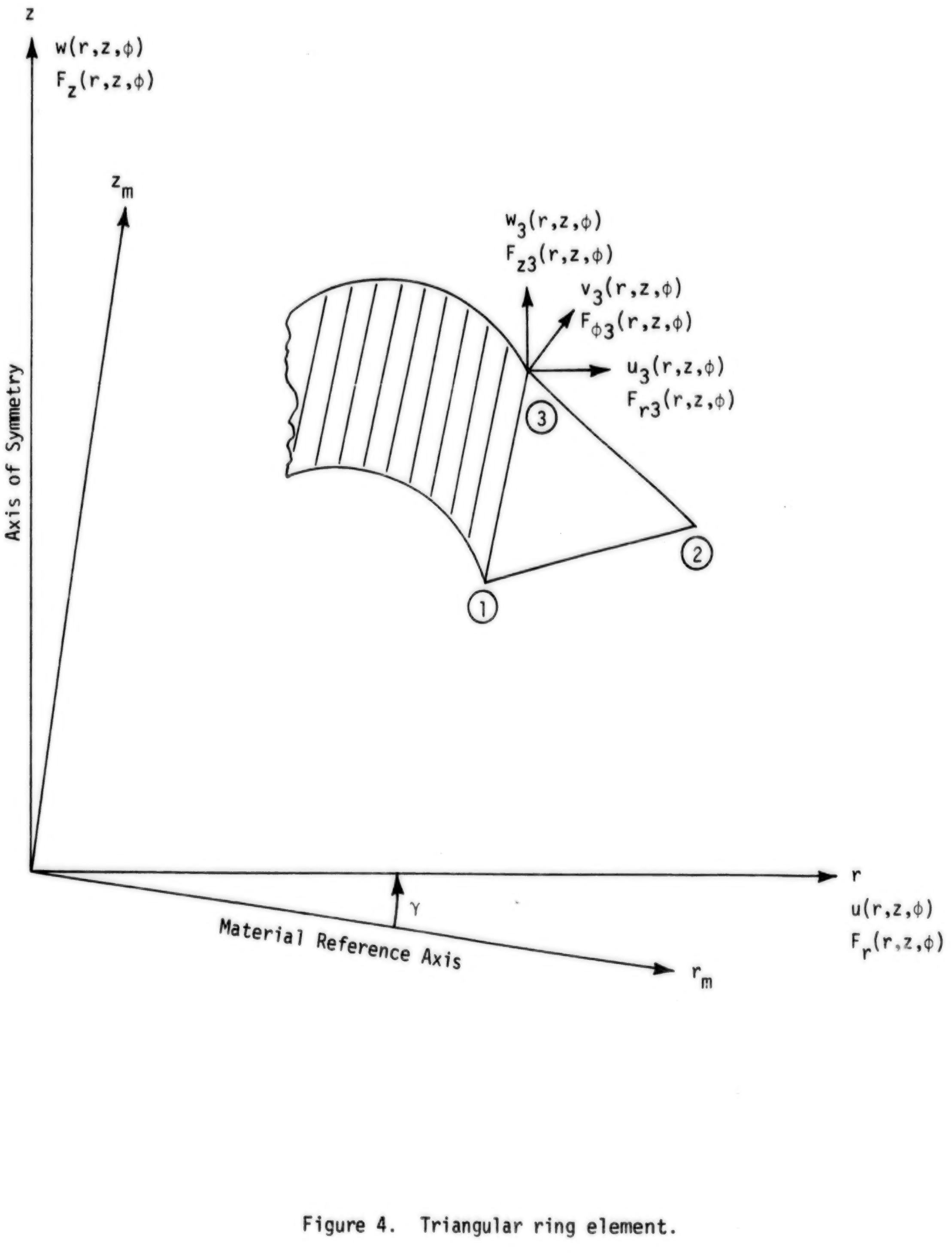

z
w(r,z,ϕ)
F_z(r,z,ϕ)
z_m
Axis of Symmetry
w_3(r,z,ϕ)
F_z3(r,z,ϕ)
v_3(r,z,ϕ)
F_ϕ3(r,z,ϕ)
u_3(r,z,ϕ)
F_r3(r,z,ϕ)
③
②
①
r
u(r,z,ϕ)
F_r(r,z,ϕ)
γ
Material Reference Axis
r_m

Figure 4. Triangular ring element.

ORIGINAL PAGE IS
OF POOR QUALITY

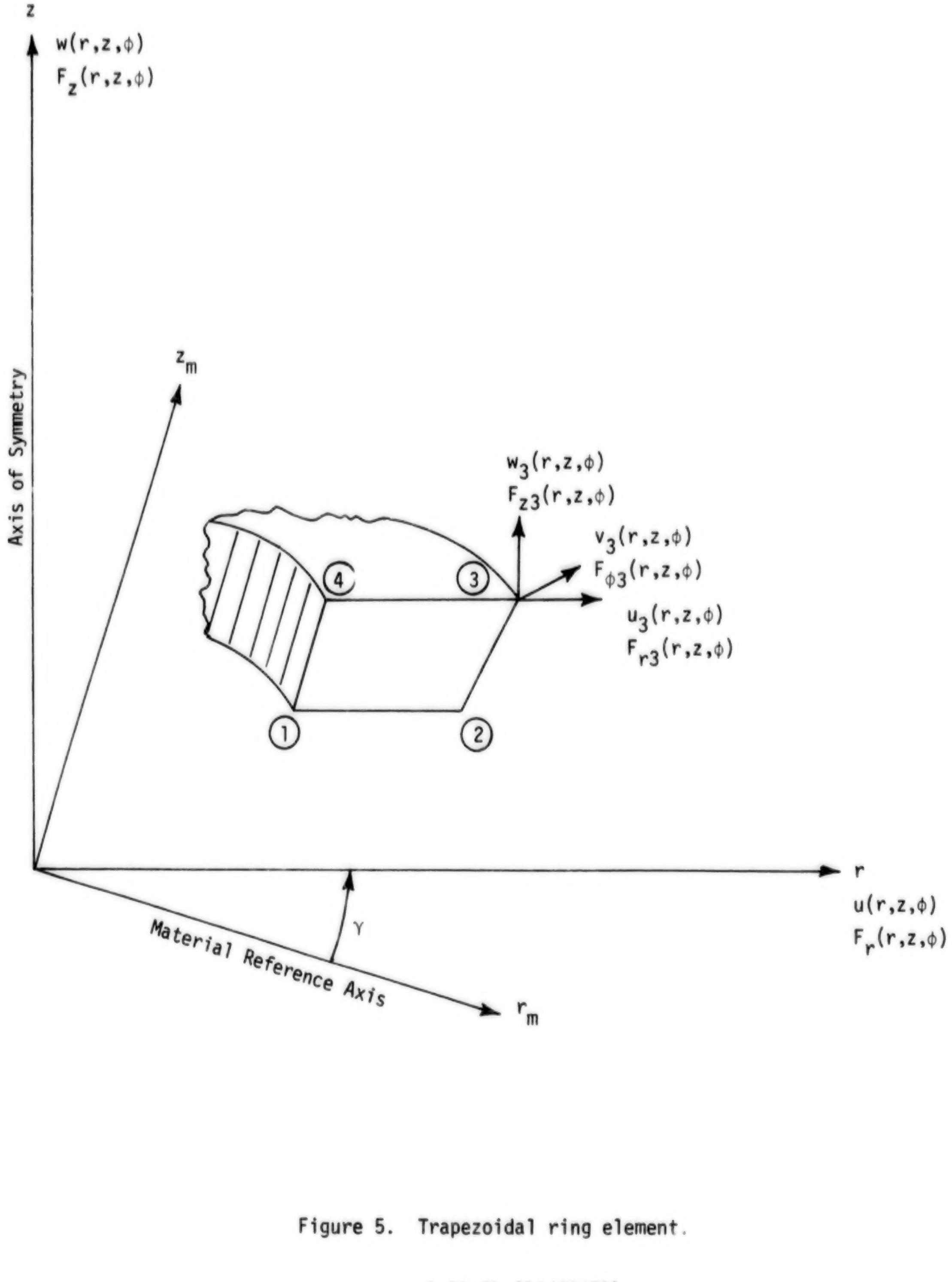

z
$w(r,z,\phi)$
$F_z(r,z,\phi)$
Axis of Symmetry
z_m
$w_3(r,z,\phi)$
$F_{z3}(r,z,\phi)$
$v_3(r,z,\phi)$
$F_{\phi 3}(r,z,\phi)$
$u_3(r,z,\phi)$
$F_{r3}(r,z,\phi)$
④
③
①
②
r
$u(r,z,\phi)$
$F_r(r,z,\phi)$
γ
Material Reference Axis
r_m

Figure 5. Trapezoidal ring element.

5.12 CONSTANT STRAIN SOLID ELEMENTS

5.12.1 Introduction

Solid polyhedron elements have been implemented to model three-dimensional elastic regions, which do not have axial symmetry (see Section 5.11 for solid of revolution elements). The geometry of the polyhedron elements is defined by grid points at the vertices. Three geometries have been implemented (see Figure 1):

1. Tetrahedron. The tetrahedron is a triangular pyramid which can be constructed between any four non-coplanar vertices. It is the basic building block which is used to build up the other elements.
2. Wedge. The wedge is a truncated triangular pyramid that is defined by six vertices. It has two triangular and three quadrilateral faces.
3. Hexahedron. The hexahedron is a generalized cube. It has six quadrilateral faces.

These elements are subject to the following conditions and limitations:

1. Constant strain in each tetrahedral subelement,
2. Uniform, isotropic material properties,
3. Uniform temperature in each tetrahedral subelement,
4. Differential stiffness, buckling and piecewise linear analysis have not been implemented,
5. Only translational degrees of freedom are used at the grid points,
6. Stress output is in the basic coordinate system.

A necessary task in formulating a finite element is to relate the coefficients of functions approximating the displacement field to those displacements at the grid points on boundaries of the element. One method to do this is to solve a set of simultaneous equations that equates the functional representation to the actual displacements at the vertices of the element. The tetrahedron element with constant strain is defined as being compatible since it has twelve grid point degrees of freedom (three translations at each of four vertices) and twelve generalized displacements (coefficients of the constant and linear terms for u, v and w). Thus, the matrix of coefficients relating the two types of displacements is nonsingular. The wedge and hexahedron are not modeled directly but are built up by tetrahedron elements. The coefficient matrices for the wedge and the hexahedron are expressed as sums of the coefficient matrices of the tetrahedra into which they are decomposed, see Figure 1.

PRECEDING PAGE BLANK NOT FILMED

5.12.2 Displacement Functions for the Tetrahedron

The displacement field is assumed to be linear in the Cartesian coordinates, x, y and z, i.e.,

$$u(x,y,z) = q_1 + q_2 x + q_3 y + q_4 z \ , \tag{1}$$

$$v(x,y,z) = q_5 + q_6 x + q_7 y + q_8 z \ , \tag{2}$$

$$w(x,y,z) = q_9 + q_{10} x + q_{11} y + q_{12} z \ . \tag{3}$$

The NASTRAN basic coordinate system will be used. The assumed displacement field produces uniform strain and stress within the element. In addition the displacements on common faces of adjoining elements are compatible. The generalized coordinates, (q's), can be determined from the displacements at the four vertices of the tetrahedron by solving:

$$\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \ , \tag{4}$$

for the u components of displacement, and similar equations for the v and w components. In Equation 4, $u_1 = u(x_1, y_1, z_1)$, etc. A similar relation is written for v_1 to v_4 and w_1 to w_4 in terms of q_5 to q_8 and q_9 to q_{12}.

The determinant of the matrix in Equation 4 is six times the volume of the tetrahedron defined by points 1, 2, 3 and 4.

$$\begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix} = \pm\, 6 \cdot \text{Volume} \ . \tag{5}$$

Hence, the matrix in Equation 4 will be nonsingular if the volume of the tetrahedron is nonzero.

5.12.3 Strain, Stress and Potential Energy

The generalized displacements are related to the grid point displacements by

$$\begin{bmatrix} q_1 & q_5 & q_9 \\ q_2 & q_6 & q_{10} \\ q_3 & q_7 & q_{11} \\ q_4 & q_8 & q_{12} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ u_3 & v_3 & w_3 \\ u_4 & v_4 & w_4 \end{bmatrix} \qquad (6)$$

In Equation 6, the $[h_{ij}]$ matrix is the inverse of the matrix of Equation 4. The equations for v and w have been adjoined as additional columns. The six strain components are given by

$$\varepsilon_x = \partial u/\partial x = q_2 \qquad (7a)$$

$$\varepsilon y = \partial v/\partial y = q_7 \qquad (7b)$$

$$\varepsilon z = \partial w/\partial z = q_{12} \qquad (7c)$$

$$\gamma_{yz} = \partial v/\partial z + \partial w/\partial y = q_8 + q_{11} \qquad (7d)$$

$$\gamma_{zx} = \partial w/\partial x + \partial u/\partial z = q_{10} + q_4 \qquad (7e)$$

$$\gamma_{xy} = \partial u/\partial y + \partial v/\partial x = q_3 + q_6 \qquad (7f)$$

Eliminating the q's from Equation 7 using Equation 6,

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} h_{21} & 0 & 0 \\ 0 & h_{31} & 0 \\ 0 & 0 & h_{41} \\ 0 & h_{41} & h_{31} \\ h_{41} & 0 & h_{21} \\ h_{31} & h_{21} & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \end{Bmatrix} + \begin{bmatrix} h_{22} & 0 & 0 \\ 0 & h_{32} & 0 \\ 0 & 0 & h_{42} \\ 0 & h_{42} & h_{32} \\ h_{42} & 0 & h_{22} \\ h_{32} & h_{22} & 0 \end{bmatrix} \begin{Bmatrix} u_2 \\ v_2 \\ w_2 \end{Bmatrix} + \begin{bmatrix} h_{23} & 0 & 0 \\ 0 & h_{33} & 0 \\ 0 & 0 & h_{43} \\ 0 & h_{43} & h_{33} \\ h_{43} & 0 & h_{23} \\ h_{33} & h_{23} & 0 \end{bmatrix} \begin{Bmatrix} u_3 \\ v_3 \\ w_3 \end{Bmatrix} + \begin{bmatrix} h_{24} & 0 & 0 \\ 0 & h_{34} & 0 \\ 0 & 0 & h_{44} \\ 0 & h_{44} & h_{34} \\ h_{44} & 0 & h_{24} \\ h_{34} & h_{24} & 0 \end{bmatrix} \begin{Bmatrix} u_4 \\ v_4 \\ w_4 \end{Bmatrix} \qquad (8)$$

This will be abbreviated as

$$\{\varepsilon\} = \sum_{i=1}^{4} [C_i]\{u\}_i \quad . \qquad (9)$$

The subscript i in $\{u\}_i$ refers to the vertex of the tetrahedron and not to the component of the vector. The elements of $[C_i]$ are either h_{2i}, h_{3i}, h_{4i}, or zero. The first row of h's is not needed. If T is the average temperature rise, the stresses are given by

$$\{\sigma\} = [G_e](\{\varepsilon\}-\{\alpha\}\bar{T}) \tag{10}$$

where the stress and thermal strain vectors are:

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{zx} \\ \tau_{xy} \end{Bmatrix} \quad \text{and} \quad \{\alpha\} = \begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{11}$$

$[G_e]$ is a (6x6) stress-strain matrix in basic coordinates. When material properties are given in terms of E and ν, then

$$\begin{gathered} G_{11} = G_{22} = G_{33} = E(1-\nu)/(1-2\nu)(1+\nu) \\ G_{12} = G_{21} = G_{13} = G_{31} = G_{23} = G_{32} = E\nu/(1-2\nu)(1+\nu) \\ G_{44} = G_{55} = G_{66} = E/2(1+\nu)\ . \end{gathered} \tag{12}$$

All other terms are zero. The strain energy density is a constant over the volume; hence, the strain energy is

$$V = (\text{Volume}) \times \frac{1}{2}\{\varepsilon\}^T\{\sigma\}\ . \tag{13}$$

The volume can be found from Equation 5.

5.12.4 Matrices in Basic Coordinates

The stiffness matrix for the tetrahedron is of order twelve. It is convenient to represent the matrix in terms of sixteen 3 x 3 matrices each representing interaction between a pair of grid points. The 3 x 3 partition of the stiffness matrix associated with grid points i and j is given by

$$[k_{ij}] = (\text{Volume})[C_i^T][G_e][C_j] \ . \tag{14}$$

The thermal loads on point i are

$$\{P^{(t)}\}_i = (\text{Volume})[C_i^T][G_e]\{\alpha\}\bar{T} \ . \tag{15}$$

One-fourth of the mass of the tetrahedron is lumped at each vertex. This choice preserves the center of gravity. The density is assumed uniform; hence, the mass at the point i is

$$m_i = \frac{1}{4}(\text{Volume})\rho \ . \tag{16}$$

5.12.5 Wedge and Hexahedron Elements

The wedge and hexahedron elements are decomposed into subtetrahedra. As seen in Figure 1, the wedge can be cut into three subtetrahedra. Of the six possible ways to subdivide a wedge, only one has been implemented. Other subdivisions and overlapping methods have not been coded since it is not expected that the wedge will be used often enough to justify the additional effort. It is expected to be used primarily as a fillet. The hexahedron can be cut into five subtetrahedra in only two different ways. The code implemented for the hexahedron allows the user to choose a single subdivision into five subtetrahedra or to use the average of the results of the two types of subdivisions. The latter choice, but not the former, results in symmetrical deformations when symmetrical loads are applied to a symmetrical hexahedron.

The geometries of the wedge and hexahedron are determined by the coordinates of the vertices. The quadrilateral faces may not be coplanar for general grid point location. There is no guarantee of the results if these are not coplanar, and the user should be careful to keep the vertices of the quadrilateral faces nearly in one plane. The resulting stiffness and thermal load matrices are the sums of those due to the subtetrahedra. All matrices must be put into global form, which requires multiplication by unitary rotation matrices.

ORIGINAL PAGE IS
OF POOR QUALITY

5.12.6 Stress Recovery

The stresses, in the basic coordinate system, can be recovered for the tetrahedron using Equations 10 and 9. In addition to the stress components, output also includes the pressure

$$p_o = -\frac{1}{3}(\sigma_x + \sigma_y + \sigma_z) \quad , \tag{17}$$

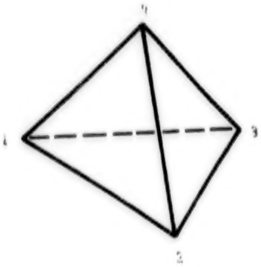
and the octahedral stress

$$\sigma_o = \frac{1}{3}\left((\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6\tau_{yz}^2 + 6\tau_{zx}^2 + 6\tau_{xy}^2\right)^{1/2} . \tag{18}$$

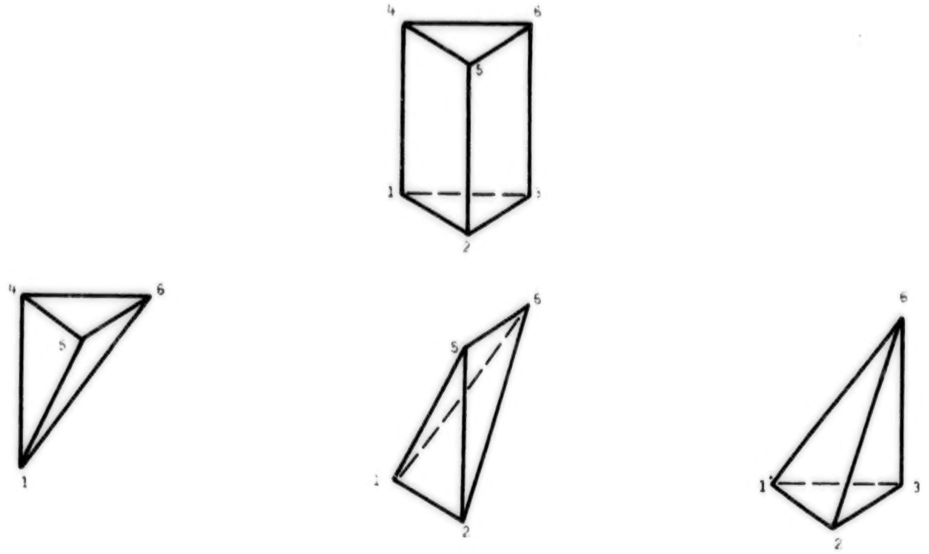
The stresses in the wedge and the hexahedron are obtained as the weighted average of the stresses in the subtetrahedra. The weighting factor for each tetrahedron is proportional to its volume.

SOLID POLYHEDRON ELEMENTS

TETRAHEDRON



WEDGE AND ONE OF ITS SIX DECOMPOSITIONS


4
6
5
1
3
2
4
6
5
1
6
5
1
2
6
1
3
2

HEXAHEDRON AND ITS TWO DECOMPOSITIONS

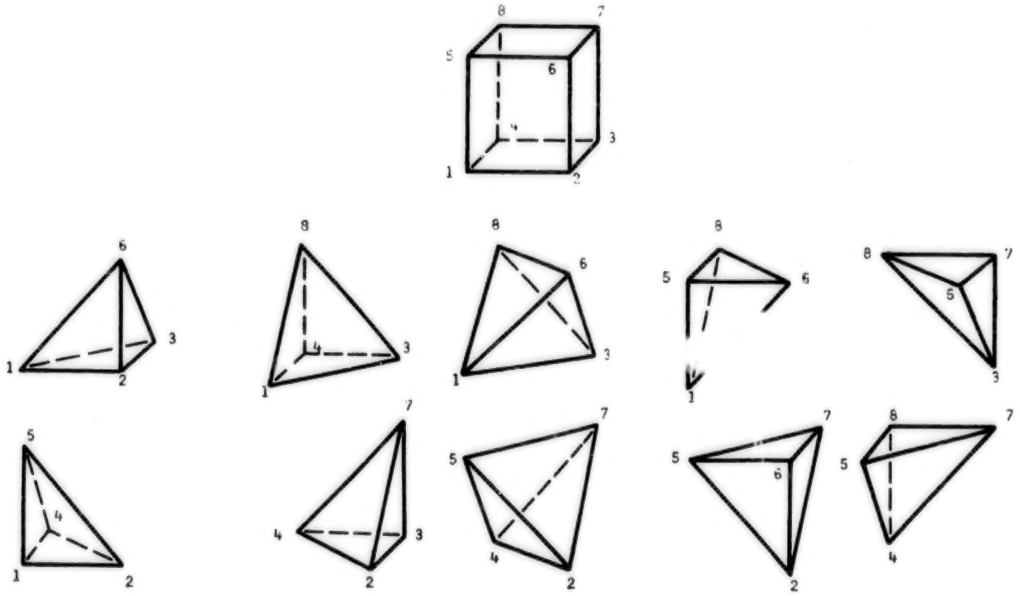

8
7
5
6
4
3
1
2
6
1
3
2
8
4
1
3
8
6
1
3
8
5
6
1
8
7
5
3
5
4
1
2
7
4
3
2
7
5
4
2
7
5
6
2
8
7
5
4

Figure 1. Polyhedron elements and their subtetrahedra.

5.13 ISOPARAMETRIC HEXAHEDRON SOLID ELEMENTS

Isoparametric hexahedron solid elements may be used to analyze any three-dimensional continuum composed of isotropic materials. Examples include thick inserts in rocket engine nozzles, thermal protection system insulations, soil structure interaction problems, and geometrically complex thick-walled mechanical components such as pumps, valves, etc. The isoparametric solid elements take into account:

1. Isotropic temperature-dependent material properties.
2. Pressure and temperature loads.
3. Coupled mass matrix.

Although solid elements employ only three degrees of freedom at each grid point (the three displacement components), they may be combined with all other NASTRAN elements except special stand-alone elements.

The isoparametric solid elements were first presented by Irons, Ergatoudis and Zienkiewicz (References 1, 2, 3 and 4). They are also called conformal higher order elements, since the displacement of the element can be represented to any degree one desires, and still maintain inter-element compatibility, by using more grid points per edge to define element geometry and deformation. In practice, however, isoparametric solid elements employing either eight, twenty, or thirty-two grid points have been found to be adequate to solve most problems (Figure 1). These elements correspond to assuming a linear, parabolic, and cubic variation of displacement, respectively. Clough (Reference 5) conducted an evaluation of three-dimensional solid elements and showed that the isoparametric elements were superior to other solid elements. He further pointed out that the choice of which isoparametric element is best to use depends on the type of problem being solved. For problems involving plate bending type deformations, the higher order elements appear to be best, while the linear element is recommended for problems in which shear stresses are likely to be large. It is for this reason that all three isoparametric elements have been incorporated into NASTRAN.

The isoparametric elements governing equations are based on minimum energy principles. The derivation of these equations consists of assuming a displacement function for the element which depends on its grid point displacements, substitute these displacements into the potential energy expression, and minimize the energy functional to obtain the governing equations. The detailed derivation is presented next.

PRECEDING PAGE BLANK NOT FILMED

PAGE 471 INTENTIONALLY BLANK

ORIGINAL PAGE IS OF POOR QUALITY

5.13.1 Displacement Functions for the Isoparametric Element

The name isoparametric comes from the fact that the same interpolating functions are used to represent both the geometry and the deformation of the element. This choice insures that the element displacement functions satisfy the criteria necessary for convergence of the finite element analysis (Reference 4). Consider the elements shown in Figure 1. The curvilinear coordinates in these elements are related to the "basic" rectangular ones by the following definition:

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \sum_{i=1}^{M} N_i(\xi, \eta, \zeta) \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_i \quad , \tag{1}$$

where M equals the total number of element grid points,

i refers to the grid point i,

and $N_i(\xi, \eta, \zeta)$ are interpolating functions which depend on the number of grid points used to define the element geometry. The N_i functions are either linear, parabolic, or cubic, and correspond to employing two, three, or four grid points, respectively, along each edge of the element. This choice insures that there are no geometric gaps between grid points. Expressions for the interpolating functions and their derivatives are presented in Table 1.

The deformations of the elements are represented with the identical interpolating functions used to define the geometry, i.e.,

$$\{\bar{u}\} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{M} N_i(\xi, \eta, \zeta) \begin{Bmatrix} u \\ v \\ w \end{Bmatrix}_i = [N]\{u_e\} \quad , \tag{2}$$

where u, v and w are displacements along the x, y and z basic coordinate axes,

and $\{u_e\}$ represents the vector of grid point displacements.

The displacement functions, $N_i(\xi, \eta, \zeta)$ satisfy the required convergence criteria of adequately representing a constant strain state, and insure interelement compatibility along the complete element boundary (Reference 4).

5.13.2 Strain-Displacement Relationship

The strains at any point within the element are given by the well known relations

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \end{Bmatrix} \quad . \tag{3}$$

The substitution of Equation 2 into Equation 3 yields the relationship for the strain vector in terms of the grid point displacements

$$\{\varepsilon\} = [C_1 \quad C_2 \quad \dots \quad C_{NGP}] \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ \vdots \\ u_{NGP} \\ v_{NGP} \\ w_{NGP} \end{Bmatrix} = [C]\{u_e\} \quad , \tag{4}$$

and

474

$$[C_i] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix} . \tag{5}$$

In order to evaluate the strain matrix [C], we must calculate the derivatives of the shape functions, N_i, with respect to x, y and z. Since N_i is defined in terms of ξ, η and ζ, it is necessary to use the relation that

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} \tag{6}$$

where [J] is the Jacobian matrix and is easily evaluated by noting that

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi}, & \frac{\partial N_2}{\partial \xi}, & \ldots, & \frac{\partial N_{NGP}}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta}, & \frac{\partial N_2}{\partial \eta}, & \ldots, & \frac{\partial N_{NGP}}{\partial \eta} \\ \frac{\partial N_1}{\partial \zeta}, & \frac{\partial N_2}{\partial \zeta}, & \ldots, & \frac{\partial N_{NGP}}{\partial \zeta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{NGP} & y_{NGP} & z_{NGP} \end{bmatrix} . \tag{7}$$

The derivatives of the shape functions with respect to ξ, η and ζ are given in Table 1, x_1, y_1 and z_1 are the coordinates of the element grid points, and NGP is the number of element grid points.

5.13.3 Stress-Strain Relations

The stress-strain relations for a general elastic anisotropic material are

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{Bmatrix} = [G_e]\{\varepsilon - \varepsilon_t\} \quad , \tag{8}$$

where for an isotropic material

$$[G_e] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad ,$$

and $\{\sigma\}$ is the stress vector in the basic rectangular Cartesian coordinate system, $[G_e]$ is, in general, a full symmetric material elastic modulus matrix, ε is the total strain vector given by Equation 3, and ε_t is the thermal strain.

$$\{\varepsilon_t\} = \{\alpha_e\}\bar{T} = \begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ \alpha_{xy} \\ \alpha_{yz} \\ \alpha_{zx} \end{Bmatrix} \bar{T} \quad , \tag{9}$$

where $\{\alpha_e\}$ is a vector of thermal expansion coefficients and $\bar{T}$ is the temperature distribution within the element, determined from user-specified grid point temperatures, $\{T_e\}$, by applying the interpolating relation

$$\bar{T} = \sum_{i=1}^{n} N_i(\xi, \eta, \zeta)T_i = [N]\{T_e\} \quad . \tag{10}$$

At present in the NASTRAN program, the user may only specify isotropic material properties. When a general anisotropic material is defined and is available to the isoparametric solid elements, then the user may specify the anisotropic material properties with respect to a particular orientation that does not necessarily coincide with the basic rectangular coordinate system. NASTRAN will then transform the material properties to the basic coordinate system by constructing a transformation matrix, U, such that the material properties in the basic coordinate system are given by

$$\left.\begin{aligned} [G_e] &= [U]^T[G_m][U] \\ \{\alpha_e\} &= [U]^{-1}\{\alpha_m\} \end{aligned}\right\} \quad , \tag{11}$$

where $[G_m]$ and $\{\alpha_m\}$ are the material modulus matrix and thermal expansion coefficient vector specified by the user.

5.13.4 Stiffness, Mass and Load Matrices

The stiffness, mass and load matrices for the isoparametric element may now be derived by application of the principle of virtual work

$$\delta U - \delta W = 0 \quad , \tag{12}$$

where δU is the internal strain energy in the element due to a virtual displacement, $\delta\bar{u}$, and δW is the work performed by the external loads during the virtual displacement, i.e.,

$$\left.\begin{aligned} \delta U &= \int_V \{\sigma\}^T\{\delta\epsilon\}dV \\ \delta W &= \int_S \{\delta\bar{u}\}^T\{p\}dS + \int_V \{\delta\bar{u}\}^T\{F\}dV \end{aligned}\right\} \quad , \tag{13}$$

where {p} and {F} are 3 x 1 vectors representing surface pressure and body forces, respectively, in the x, y and z directions, and the integrations are performed over the element volume, V, and surface area, S, on which the pressure load, p, acts.

ORIGINAL PAGE IS
OF POOR QUALITY

Substituting Equations 2, 4 and 8 into Equation 13 and applying the principle of virtual work, Equation 12, yields

$$\{\delta u_e\}^T([K_{gg}]\{u_e\} - \{F_e\}) = 0 \quad , \tag{14}$$

or, for any virtual displacement

$$[K_{gg}]\{u_e\} = \{F_e\} \quad , \tag{15}$$

where $[K_{gg}]$ is the element stiffness matrix and $\{F_e\}$ is the element load vector to surface pressures and temperature.

Stiffness Matrix

$$[K_{gg}] = \int [C]^T[G_e][C]dV \quad , \tag{16}$$

where the infinitesimal volume is in terms of the curvilinear coordinates, ξ, η and ζ,

$$dV = dx\ dy\ dz = \det.[J]d\xi\ d\eta\ d\zeta \quad . \tag{17}$$

Surface Pressure Load

$$\{F_p\} = \int_S [N]^T\{p\}dS \quad . \tag{18}$$

Thermal Load

$$\{F_T\} = \int_V [C]^T[G_e]\{\alpha_e\}[N]\{T_e\}dV \quad . \tag{19}$$

For dynamics problems, the mass matrix is also required and is easily derived by adding the kinetic energy to Equation 12. The result is:

Mass Matrix

$$[M] = \int_V [N]^T[N]\rho dV \quad , \tag{20}$$

where ρ is the mass density.

478

5.13.5 Numerical Integration

The integrals in the isoparametric element stiffness, mass and load matrices are evaluated by the use of numerical integration. The Gaussian Quadrature Formula (Reference 6) is used in NASTRAN. Thus, for example, the stiffness matrix is calculated by the triple summation

$$[K] = \int_{-1}^{1} \int_{-1}^{1} \int_{-1}^{1} [C]^T[G_e][C] \; |J| \; d\xi \, d\eta \, d\zeta = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} H_i \; H_j \; H_k([C]^T[G_e][C]) \; |J| \; , \quad (21)$$

where the weight coefficients H_ℓ and abscissa S_ℓ are given in Table 2. Note that the triple product matrix operation in Equation 16 as well as the determinant of the Jacobian, $|J|$, must be evaluated at each integration point. This process could be very time consuming, and requires that efficient programming practices and mathematical techniques be used to minimize this time. In NASTRAN, for isotropic materials, the triple product in Equation 21 is explicitly evaluated to avoid calculating zeroes and thereby minimizes the number of mathematical operations performed.

The number of integration points needed to evaluate the stiffness, mass and load matrices depends on the element geometry, displacement function and material property variations. Elements which are extremely distorted from a rectangular shape require more integration points. Best results, however, are obtained using rectangular elements as far as possible, and, therefore, extremely distorted elements should be avoided (References 8 and 9). It has been found that for most problems satisfactory results may be obtained using a 2 x 2 x 2 integration mesh for the linear element and a 3 x 3 x 3 integration mesh for the quadratic and cubic elements. These meshes are used as default values in NASTRAN. However, since good results have also been reported using smaller meshes than suggested above (References 7 and 8), the user has the option of specifying the integration mesh size.

5.13.6 Transformation from Basic to Global Coordinates

As previously stated, all computations for the isoparametric elements are carried out in the basic coordinate system. If the global coordinate system at any grid point is different from the basic system, NASTRAN transforms the final matrices and vectors into the global coordinate system using the appropriate transformation matrix, $[T_i]$, corresponding to grid point i. This calculation is identical to that performed for the other elements in NASTRAN.

5.13.7 Stress Recovery

Element stresses may be obtained by combining Equations 4, 8 and 9 to yield

$$\{\sigma\} = [G_e]([C]\{u_e\} - \{\alpha_e\}[N]\{T_e\}) \quad , \tag{22}$$

where the matrices [C] and [N] are functions of the element coordinates ξ, η, and ζ. In NASTRAN these stresses are calculated in the basic coordinate system and they are printed at the following locations, depending on element type,

a. Linear Element - Eight corner points and at center of element.

b. Quadratic and Cubic Element - Eight corner points, center of each edge, and at center of element.

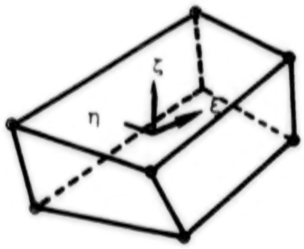
The principal stresses, principal angles, mean stress and octahedral shear stress are also computed and output at every point at which the basic stresses are computed. The mean stress, or hydrostatic pressure, is given by

$$\sigma_n = -\frac{1}{3}(\sigma_x + \sigma_y + \sigma_z) \quad . \tag{23}$$

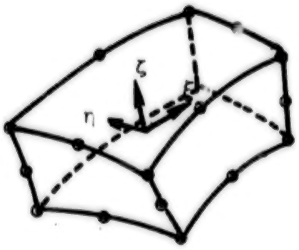
The octahedral shear stress is given by

$$\sigma_o = \left\{\frac{1}{3}\left[(S_x + \sigma_n)^2 + (S_y + \sigma_n)^2 + (S_z + \sigma_n)^2\right]\right\}^{1/2} \quad , \tag{24}$$

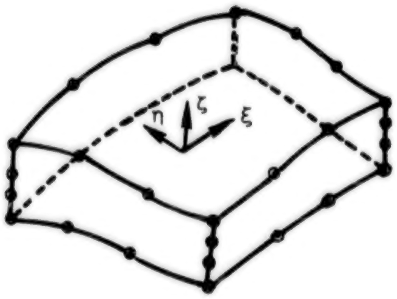
where S_x, S_y and S_z are the three principal stresses.



(a) Linear



(b) Quadratic



(c) Cubic

Figure 1. Three isoparametric elements

Table 1a. Isoparametric shape functions and their derivatives for linear element - 8 grid points

CORNER GRID POINTS

$$\xi_i = \pm 1\,,\ \eta_i = \pm 1\,,\ \zeta_i = \pm 1$$

$$N_i = \frac{1}{8}(1 + \xi_o)\ (1 + \eta_o)\ (1 + \zeta_o)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{1}{8}\,\xi_i\ (1 + \eta_o)\ (1 + \zeta_o)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{1}{8}\,\eta_i\ (1 + \xi_o)\ (1 + \zeta_o)$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{1}{8}\,\zeta_i\ (1 + \xi_o)\ (1 + \eta_o)$$

where $\xi_o = \xi\xi_i$,

$\eta_o = \eta\eta_i$,

$\zeta_o = \zeta\zeta_i$.

Table 1b. Isoparametric shape functions and their derivatives for parabolic element - 20 grid points

CORNER GRID POINTS

$$\xi_i = \pm 1 \,, \ \eta_i = \pm 1 \,, \ \zeta_i = \pm 1$$

$$N_i = \frac{1}{8}(1 + \xi_o)(1 + \eta_o)(1 + \zeta_o)(\xi_o + \eta_o + \zeta_o - 2)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{1}{8}\xi_i \ (1 + \eta_o)(1 + \zeta_o)(2\xi_o + \eta_o + \zeta_o - 1)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{1}{8}\eta_i \ (1 + \xi_o)(1 + \zeta_o)(2\eta_o + \xi_o + \zeta_o - 1)$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{1}{8}\zeta_i \ (1 + \zeta_o)(1 + \xi_o)(2\zeta_o + \xi_o + \eta_o - 1)$$

MID-SIDE GRID POINTS

$$\xi_i = 0, \ \eta_i = \pm 1 \,, \ \zeta_i = \pm 1$$

$$N_i = \frac{1}{4}(1 - \xi^2)(1 + \eta_o)(1 + \zeta_o)$$

$$\frac{\partial N_i}{\partial \xi} = -\frac{1}{2}\xi(1 + \eta_o)(1 + \zeta_o)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{1}{4}(1 - \xi^2)(1 + \zeta_o)\eta_i$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{1}{4}(1 - \xi^2)(1 + \eta_o)\zeta_i$$

MID-SIDE GRID POINTS

$$\xi_i = \pm 1 \,, \ \eta_i = 0, \ \zeta_i = \pm 1$$

$$N_i = \frac{1}{4}(1 - \eta^2)(1 + \xi_o)(1 + \zeta_o)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{1}{4}(1 - \eta^2)(1 + \zeta_o)\xi_i$$

$$\frac{\partial N_i}{\partial \eta} = -\frac{1}{2}\eta(1 + \zeta_o)(1 + \xi_o)$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{1}{4}(1 - \eta^2)(1 + \xi_o)\zeta_i$$

MID-SIDE GRID POINTS

$$\xi_i = \pm 1 \,, \ \eta_i = \pm 1 \,, \ \zeta_i = 0$$

$$N_i = \frac{1}{4}(1 - \zeta^2)(1 + \xi_o)(1 + \eta_o)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{1}{4}(1 - \zeta^2)(1 + \eta_o)\xi_i$$

$$\frac{\partial N_i}{\partial \eta} = \frac{1}{4}(1 - \zeta^2)(1 + \xi_o)\eta_i$$

$$\frac{\partial N_i}{\partial \zeta} = -\frac{1}{2}\zeta(1 + \xi_o)(1 + \eta_o)$$

where $\xi_o = \xi\xi_i$,

$\eta_o = \eta\eta_i$,

$\zeta_o = \zeta\zeta_i$.

Table 1c. Isoparametric shape functions and their derivatives for cubic element - 32 grid points

CORNER GRID POINTS

$\xi_i = \pm 1, \eta_i = \pm 1, \zeta_i = \pm 1$

$$N_i = \frac{9}{64}(1+\xi_o)(1+\eta_o)(1+\zeta_o)(\xi^2+\eta^2+\zeta^2-19)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{9}{64}(1+\eta_o)(1+\zeta_o)[\xi_i(3\xi^2+\eta^2+\zeta^2-19)+2\xi]$$

$$\frac{\partial N_i}{\partial \eta} = \frac{9}{64}(1+\xi_o)(1+\zeta_o)[\eta_i(3\eta^2+\xi^2+\zeta^2-19)+2\eta]$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{9}{64}(1+\xi_o)(1+\eta_o)[\zeta_i(3\zeta^2+\xi^2+\eta^2-19)+2\zeta]$$

MID-SIDE GRID POINTS

$\xi_i = \pm \frac{1}{3}, \eta_i = \pm 1, \zeta_i = \pm 1$

$$N_i = \frac{9}{64}(1-\xi^2)(1+9\xi_o)(1+\eta_o)(1+\zeta_o)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{9}{64}(1+\eta_o)(1+\zeta_o)(-2\xi+9\xi_i-18\xi\xi_o)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{9}{64}(1-\xi^2)(1+9\xi_o)(1+\zeta_o)\eta_i$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{9}{64}(1-\xi^2)(1+9\xi_o)(1+\eta_o)\zeta_i$$

MID-SIDE GRID POINTS

$\xi_i = \pm 1, \eta_i = \pm \frac{1}{3}, \zeta_i = \pm 1$

$$N_i = \frac{9}{64}(1-\eta^2)(1+9\eta_o)(1+\xi_o)(1+\zeta_o)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{9}{64}(1-\eta^2)(1+9\eta_o)\xi_i$$

$$\frac{\partial N_i}{\partial \eta} = \frac{9}{64}(1+\xi_o)(1+\zeta_o)(-2\eta+9\eta_i-18\eta\eta_o)$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{9}{64}(1-\eta^2)(1+9\eta_o)\zeta_i$$

MID-SIDE GRID POINTS

$\xi_i = \pm 1, \eta_i = \pm 1, \zeta_i = \pm \frac{1}{3}$

$$N_i = \frac{9}{64}(1-\zeta^2)(1+9\zeta_o)(1+\xi_o)(1+\eta_o)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{9}{64}(1-\zeta^2)(1+9\zeta_o)\eta_i$$

$$\frac{\partial N_i}{\partial \xi} = \frac{9}{64}(1-\zeta^2)(1+9\zeta_o)\xi_i$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{9}{64}(1+\xi_o)(1+\eta_o)(-2\zeta+9\zeta_i-18\zeta\zeta_i)$$

where $\xi_o = \xi\xi_i$, $\eta_o = \eta\eta_i$, $\zeta_o = \zeta\zeta_i$.

Table 2. Gaussian quadrature formula

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(x, y, z)dx\, dy\, dz = \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n} H_j\, f(s_j, s_k, s_i)$$

n	Abscissa (s)	Weight Coefficient (H)
2	±0.57735026919	1.0
3	±0.77459666924	0.55555555555
	0.0	0.88888888888
4	±0.86113631159	0.34785484514
	±0.33998104358	0.65214515486

REFERENCES

1. Irons, B. M., "Engineering Application of Numerical Integration in Stiffness Methods," AIAA J., Vol. 4, pp. 2035-2037, November 1966.

2. Irons, B. M., and Zienkiewicz, O. C., "The Iso-Parametric Element System - A New Conecpt in Finite Element Analysis," Conf. Recent Advances in Stress Analysis, J.B.C.S.A., Royal Aero. Soc., London, 1968.

3. Ergatoudis, I., "Quadrilateral Elements in Plane Analysis and Introduction to Solid Analysis," M.Sc. Thesis, University of Wales, Swansea, 1966.

4. Ergatoudis, I., Irons, B. M., and Zienkiewicz, O. C., "Curved, Isoparametric, 'Quadrilateral' Elements for Finite Element Analysis," Int. J. Solids Struct., Vol. 4, 1968, pp. 31-42.

5. Clough, R. W., "Comparison of Three Dimensional Finite Elements," Symposium on Appl. Finite Element Method in Civil Engineering, Vanderbilt University, Nashville, Tenn., November 1969.

6. Kopal, Z., Numerical Analysis, 2nd Edition, Chapman and Hill, 1961.

7. Zienkiewicz, O. C., Taylor, R. L., and Too, J. M., "Reduced Integration Technique in General Analysis of Plates and Shells," Int. J. Num. Meth. Engineering, Vol. 3, 1971, pp. 275-291.

8. Pawsey, S. R., and Clough, R. W., "Improved Numerical Integration of Thick Shell Finite Elements," Int. J. Num. Meth. Engineering, Vol. 3, 1971, pp. 575-586.

9. Jordan, W. B., The Plane Isoparametric Structural Element, Knolls Atomic Power Laboratory Report KAPL M-7112, February 1970.

5.14 THE TRIANGULAR THIN SHALLOW SHELL ELEMENT

5.14.1 Introduction

This higher order element (TRSHL) was developed by Narayanaswami (Reference 1). The element has grid points at the vertices and at the midpoints of the sides of the triangle. At each grid point, there are five degrees of freedom in the element coordinate system, viz., the membrane displacements u and v parallel to the x and y axes, the transverse displacement, w, in the z-direction normal to the x-y plane, with positive direction outward from the paper, and the rotations of the normal to the shell, α and β, about the y-z and x-z planes, with positive directions following from the right-hand rule. The element, thus, has 30 degrees of freedom in the element coordinate system.

The membrane displacements u and v for the shell are expressed as quadratic polynomials and are the same as for the linear strain triangular membrane element, TRIM6. The displacement function for the normal deflection, w, is taken as a quintic polynomial as for the higher order triangular plate bending element, TRPLT1. The geometry of the shell surface is approximated by a quadratic polynomial in the coordinates of the base. The shallow shell theory of Novozhilov (Reference 2) is used for including the membrane-bending coupling effects. Thus, the element can strictly be used only in cases where the shell is shallow. However, reasonably good accuracy is seen even when the elements are used to analyze shells that are only marginally shallow. The user is cautioned, however, to be careful while interpreting results obtained when the shell analyzed is deep. Due to the excessive computation time associated with such calculations, the transverse shear flexibility is not taken into account in the element formulation. The element can be used in the statics, normal modes and differential stiffness rigid formats.

5.14.2 Element Geometry

Rectangular Cartesian coordinates are used in the formulation. An arbitrary triangular element is shown in Figure 1. X, Y, and Z are the basic coordinates; x, y, and z are the local coordinates. The grid points of the element are numbered in counterclockwise direction as shown in the figure.

5.14.3 Displacement Field

The displacements u(x,y) and v(x,y) at any point (x,y) in the element are assumed to vary quadratically with position on the plane of the element, while the displacement w(x,y) within the

PAGE 487 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

triangular element is assumed to vary as a quintic polynomial in the local coordinates. Thus,

$$\left.\begin{aligned} u(x,y) &= a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 \quad , \\ v(x,y) &= a_7 + a_8x + a_9y + a_{10}x^2 + a_{11}xy + a_{12}y^2 \quad , \\ \text{and} \qquad w(x,y) &= a_{13} + a_{14}x + a_{15}y + a_{16}x^2 + a_{17}xy + a_{18}y^2 \\ &\quad + a_{19}x^3 + a_{20}x^2y + a_{21}xy^2 + a_{22}y^3 + a_{23}x^4 \\ &\quad + a_{24}x^3y + a_{25}x^2y^2 + a_{26}xy^3 + a_{27}y^4 + a_{28}x^5 \\ &\quad + a_{29}x^4y + a_{30}x^3y^2 + a_{31}x^2y^3 + a_{32}xy^4 + a_{33}y^5 \quad . \end{aligned}\right\} \quad . \qquad (1)$$

In concise form, u, v, and w can be written as

$$u = \sum_{i=1}^{33} a_i\, x^{m_i}\, y^{n_i} \; , \qquad a_i = m_i = n_i = 0, \; i = 7 \text{ to } 33 \quad , \qquad (2)$$

$$v = \sum_{i=1}^{33} b_i\, x^{p_i}\, y^{q_i} \; , \qquad b_i = p_i = q_i = 0, \; i = 1 \text{ to } 6 \text{ and } i = 13 \text{ to } 33 \quad , \qquad (3)$$

and

$$w = \sum_{i=1}^{33} c_i\, x^{r_i}\, y^{s_i} \; , \qquad c_i = r_i = s_i = 0, \; i = 1 \text{ to } 12 \quad . \qquad (4)$$

5.14.4 Derivation of the Stiffness Matrix

The detailed derivation of the stiffness matrix for the triangular shell element follows closely that for the TRIM6 and TRPLT1 elements. Hence, only the salient features of the derivation are given in this section.

The geometry of the shell surface is approximated by a quadratic polynomial in the coordinates of the base:

$$z(x,y) = h_1 + h_2x + h_3y + h_4x^2 + h_5xy + h_6y^2 \quad . \qquad (5)$$

Hence the curvatures of the shell surface are

$$z,_{xx} = 2h_4 \quad , \tag{6}$$

$$z,_{xy} = h_5 \quad , \tag{7}$$

and

$$z,_{yy} = 2h_6 \quad . \tag{8}$$

The membrane thickness of the shell element is assumed to vary linearly over the surface of the element, i.e.,

$$t_m = \sum_{i=1}^{3} d_i \, x^{t_i} \, y^{u_i} \quad . \tag{9}$$

The bending thickness of the shell element is also assumed to have a similar linear variation:

$$t_b = \sum_{i=1}^{3} d_i' \, x^{t_i'} \, y^{u_i'} \quad . \tag{10}$$

Following the shallow shell theory of Novoshilov (Reference 2), the membrane strains in the shell are given by

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} - z,_{xx} \, w \\ &= \sum_{i=1}^{33} \left(m_i \, a_i \, x^{m_i - 1} \, y^{n_i} - 2h_4 \, c_i \, x^{r_i} \, y^{s_i} \right) \quad , \end{aligned} \tag{11}$$

$$\begin{aligned} \varepsilon_y &= \frac{\partial v}{\partial y} - z,_{yy} \, w \\ &= \sum_{i=1}^{33} \left(q_i \, b_i \, x^{p_i} \, y^{q_i - 1} - 2h_6 \, c_i \, x^{r_i} \, y^{s_i} \right) \quad , \end{aligned} \tag{12}$$

and

$$\begin{aligned} \varepsilon_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} - 2z,_{xy} \, w \\ &= \sum_{i=1}^{33} \left(n_i \, a_i \, x^{m_i} \, y^{n_i - 1} + p_i \, b_i \, x^{p_i - 1} \, y^{q_i} \right. \\ &\qquad \left. - 2h_5 \, c_i \, x^{r_i} \, y^{s_i} \right) \quad . \end{aligned} \tag{13}$$

In the absence of transverse shear effects, the bending strains are given by

$$\chi_x = \frac{\partial^2 w}{\partial x^2} = \sum_{i=1}^{33} r_i\,(r_i - 1)\,c_i\,x^{r_i-2}\,y^{s_i} \quad , \tag{14}$$

$$\chi_y = \frac{\partial^2 w}{\partial y^2} = \sum_{i=1}^{33} s_i\,(s_i - 1)\,c_i\,x^{r_i}\,y^{s_i-2} \quad , \tag{15}$$

and

$$\chi_{xy} = 2\,\frac{\partial^2 w}{\partial x \partial y} = \sum_{i=1}^{33} 2\,r_i\,s_i\,c_i\,x^{r_i-1}\,y^{s_i-1} \quad . \tag{16}$$

Following the procedure outlined in Sections 5.8.6 and 5.8.7, the j^th^ column of the i^th^ row of the generalized stiffness matrix is obtained as

$$\begin{aligned}
k_{ij} = \sum_{k=1}^{3} \Big[G_{11}\Big(&m_i\,m_j\,d_k\,F(m_i + m_j + t_k - 2,\ n_i + n_j + u_k) \\
&- h_4\,m_i\,d_k\,F(m_i + r_j + t_k - 1,\ n_i + s_j + u_k) \\
&- h_4\,m_j\,d_k\,F(m_j + r_i + t_k - 1,\ n_j + s_i + u_k) \\
&+ h_4^2\,d_k\,F(r_i + r_j + t_k,\ s_i + s_j + u_k)\Big) \\
+ G_{22}\Big(&q_i\,q_j\,d_k\,F(p_i + p_j + t_k,\ q_i + q_j + u_k - 2) \\
&- h_6\,q_i\,d_k\,F(p_i + r_j + t_k,\ q_i + s_j + u_k - 1) \\
&- h_6\,q_j\,d_k\,F(r_i + p_j + t_k,\ s_i + q_j + u_k - 1) \\
&+ h_6^2\,d_k\,F(r_i + r_j + t_k,\ s_i + s_j + u_k)\Big) \\
+ G_{33}\Big(&n_i\,n_j\,d_k\,F(m_i + m_j + t_k,\ n_i + n_j + u_k - 2) \\
&+ n_i\,p_j\,d_k\,F(m_i + p_j + t_k - 1,\ n_i + q_j + u_k - 1) \\
&- h_5\,n_i\,d_k\,F(m_i + r_j + t_k,\ n_i + s_j + u_k - 1) \\
&+ p_i\,n_j\,d_k\,F(p_i + m_j + t_k - 1,\ q_i + n_j + u_k - 1) \\
&+ p_i\,p_j\,d_k\,F(p_i + p_j + t_k - 2,\ q_i + q_j + u_k\ )
\end{aligned} \tag{17}$$

(continued)

$$- h_5\, p_i\, d_k\, F(p_i + r_j + t_k - 1,\; q_i + s_j + u_k)$$

$$- h_5\, n_j\, d_k\, F(r_i + m_j + t_k,\; s_i + n_j + u_k - 1)$$

$$- h_5\, p_j\, d_k\, F(r_i + p_j + t_k - 1,\; s_i + q_j + u_k)$$

$$+ h_5^2\, d_k\, F(r_i + r_j + t_k,\; s_i + s_j + u_k)\Big)$$

$$+ G_{12}\Big(m_i\, q_j\, d_k\, F(m_i + p_j + t_k - 1,\; n_i + q_j + u_k - 1)$$

$$- h_6\, m_i\, d_k\, F(m_i + r_j + t_k - 1,\; n_i + s_j + u_k)$$

$$- h_4\, q_j\, d_k\, F(r_i + p_j + t_k,\; s_i + q_j + u_k - 1)$$

$$+ 2h_4\, h_6\, d_k\, F(r_i + r_j + t_k,\; s_i + s_j + u_k)$$

$$- q_i\, m_j\, d_k\, F(p_i + m_j + t_k - 1,\; q_i + n_j + u_k - 1)$$

$$- h_4\, q_i\, d_k\, F(p_i + r_j + t_k,\; q_i + s_j + u_k - 1)$$

$$- h_6\, m_j\, d_k\, F(r_i + m_j + t_k - 1,\; s_i + n_j + u_k)\Big)$$

$$+ G_{13}\Big(m_i\, n_j\, d_k\, F(m_i + m_j + t_k - 1,\; n_i + n_j + u_k - 1)$$

$$+ m_i\, p_j\, d_k\, F(m_i + p_j + t_k - 2,\; n_i + q_j + u_k)$$

$$- h_5\, m_i\, d_k\, F(m_i + r_j + t_k - 1,\; n_i + s_j + u_k)$$

$$- h_4\, n_j\, d_k\, F(r_i + m_j + t_k,\; s_i + n_j + u_k - 1)$$

$$- h_4\, p_j\, d_k\, F(r_i + p_j + t_k - 1,\; s_i + q_j + u_k)$$

$$+ 2h_4\, h_5\, d_k\, F(r_i + r_j + t_k,\; s_i + s_j + u_k)$$

$$+ n_i\, m_j\, d_k\, F(m_i + m_j + t_k - 1,\; n_i \quad n_j + u_k - 1)$$

$$- h_4\, n_i\, d_k\, F(m_i + r_j + t_k,\; n_i + s_j + u_k - 1)$$

$$+ p_i\, m_j\, d_k\, F(p_i + m_j + t_k - 2,\; q_i + n_j + u_k)$$

$$- h_4\, p_i\, d_k\, F(p_i + r_j + t_k - 1,\; q_i + s_j + u_k) \qquad (17)$$

(continued)

$$
\begin{aligned}
&- h_5\, m_j\, d_k\, F(r_i + m_j + t_k - 1,\, s_i + n_j + u_k)\Big) \\
&+ G_{23} \Big(q_i\, n_j\, d_k\, F(p_i + m_j + t_k,\, q_i + n_j + u_k - 2) \\
&\quad + q_i\, p_j\, d_k\, F(p_i + p_j + t_k - 1,\, q_i + q_j + u_k - 1) \\
&\quad - h_5\, q_i\, d_k\, F(p_i + r_j + t_k,\, q_i + s_j + u_k - 1) \\
&\quad - h_6\, n_j\, d_k\, F(r_i + m_j + t_k,\, s_i + n_j + u_k - 1) \\
&\quad - h_6\, p_j\, d_k\, F(r_i + p_j + t_k - 1,\, s_i + q_j + u_k) \\
&\quad + 2h_5\, h_6\, d_k\, F(r_i + r_j + t_k,\, s_i + s_j + u_k) \\
&\quad + n_i\, q_j\, d_k\, F(m_i + p_j + t_k,\, n_i + q_j + u_k - 2) \\
&\quad - h_6\, n_i\, d_k\, F(m_i + r_j + t_k,\, n_i + s_j + u_k - 1) \\
&\quad + p_i\, q_j\, d_k\, F(p_i + p_j + t_k - 1,\, q_i + q_j + u_k - 1) \\
&\quad - h_6\, p_i\, d_k\, F(p_i + r_j + t_k - 1,\, q_i + s_j + u_k) \\
&\quad - h_5\, q_j\, d_k\, F(r_i + p_j + t_k,\, s_i + q_j + u_k - 1)\Big) \Big] \\
&+ \sum_{k_1=1}^{3} \sum_{k_2=1}^{3} \sum_{k_3=1}^{3} \Big[\frac{1}{12}\, d'_{k_1} d'_{k_2} d'_{k_3} \big(G_{11}\, r_i\, r_j\, (r_i - 1)(r_j - 1) \\
&\bullet F(r_i + r_j + t'_{k_1} + t'_{k_2} + t'_{k_3} - 4,\, s_i + s_j + u'_{k_1} + u'_{k_2} + u'_{k_3}) \\
&+ G_{22}\, s_i\, s_j\, (s_i - 1)(s_j - 1)\, F(r_i + r_j + t'_{k_1} + t'_{k_2} + t'_{k_3}, \\
&\quad s_i + s_j + u'_{k_1} + u'_{k_2} + u'_{k_3} - 4) \\
&\quad + \Big(4G_{33}\, r_i\, r_j\, s_i\, s_j + G_{12}\{r_i\, s_j\, (r_i - 1)(s_j - 1) \\
&\quad + r_j\, s_i\, (r_j - 1)(s_i - 1)\}\Big)\; F(r_i + r_j + t'_{k_1} + t'_{k_2} + t'_{k_3} - 2, \\
&\quad s_i + s_j + u'_{k_1} + u'_{k_2} + u'_{k_3} - 2) \\
&+ 2G_{13}\{r_i\, r_j\, s_j\, (r_i - 1) + r_i\, r_j\, s_i\, (r_j - 1)\}\, F(r_i + r_j
\end{aligned}
\qquad (17)
$$

(continued)

$$+ t'_{k_1} + t'_{k_2} + t'_{k_3} - 3,\ s_i + s_j + u'_{k_1} + u'_{k_2} + u'_{k_3} - 1)$$

$$+ 2G_{23}\{r_j\ s_i\ s_j\ (s_i - 1) + r_i\ s_i\ s_j\ (s_j - 1)\}\ F(r_i + r_j$$

$$+ t'_{k_1} + t'_{k_2} + t'_{k_3} - 1,\ s_i + s_j + u'_{k_1} + u'_{k_2} + u'_{k_3} - 3)] \quad . \qquad (17)$$

(concluded)

The generalized stiffness matrix can be transformed to the element and global coordinates by transformations similar to those for the TRIM6 and TRPLT1 elements.

5.14.5 Equivalent Thermal Load Vector

The equivalent thermal load vector for the triangular thin shallow shell element consists of loads due to thermal expansion as well as due to thermal bending caused by variation of temperature with depth. The detailed derivation of the thermal load vector is similar to that developed for the TRIM6 and TRPLT1 elements; hence, only the essential steps are given here.

The vector of thermal strains is

$$\{\varepsilon_t\} = \begin{Bmatrix} \varepsilon_{xt} \\ \varepsilon_{yt} \\ \varepsilon_{xyt} \end{Bmatrix} = \begin{Bmatrix} \alpha_{e_1} \\ \alpha_{e_2} \\ \alpha_{e_{12}} \end{Bmatrix} (\bar{T} - T_{ref}) = \{\alpha_e\}\ (\bar{T} - T_{ref}) \quad , \qquad (18)$$

where $\{\alpha_e\} = [U]^{-1}\ \{\alpha_m\}$ is a vector of thermal expansion coefficients, $[U]$ is the strain transformation matrix given by Equation 15 in Section 5.8.1, $\{\alpha_m\}$ is the vector of thermal coefficients in the material axis system, T_{ref} is the reference or stress-free temperature of the material and $\bar{T}$ is the temperature at any point (x,y) in the element.

An applied stress vector which would produce the equivalent thermal strains is

$$\{\varepsilon_t\} = [G_e]\ \{\varepsilon_t\} = [G_e]\ \{\alpha_e\}\ (\bar{T} - T_{ref}) \quad . \qquad (19)$$

The generalized equivalent thermal load vector $\{P^t_{gen}\}$ is obtained as

$$\{P^t_{gen}\} = \frac{\partial}{\partial\{a\}} \int_v \{\varepsilon\}^t\ \{\sigma_t\}\ dv \quad . \qquad (20)$$

The strain vector $\{\varepsilon\}$ is given by

$$\{\varepsilon\} = \left\{\begin{matrix}\varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy}\end{matrix}\right\} = \left\{\begin{matrix}\frac{\partial u}{\partial x} - z,_{xx}\, w - z\, \chi_x \\ \frac{\partial v}{\partial y} - z,_{yy}\, w - z\, \chi_y \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} - 2z,_{xy}\, w - z\, \chi_{xy}\end{matrix}\right\} \quad , \tag{21}$$

where $z,_{xx}$, $z,_{yy}$ and $z,_{xy}$ are the curvatures of the shell surface and z is measured from the neutral surface of the element.

The temperature $\overline{T}$ at any point (x,y,z) is given by

$$\overline{T} = T_o + T'\, z \quad , \tag{22}$$

where T_o is the mean temperature and T' is the thermal gradient.

The following derivation to obtain the equivalent thermal load vector is given for the case of linear variation of thermal gradient over the planar coordinates of the element; the values of the thermal gradient at the three vertices being defined as T_1', T_3', and T_5'. This capability is not operational in NASTRAN currently. The derivation, however, is valid for cases with the same thermal gradient at the three vertices by setting T_3' and T_5' equal to T_1'. Thus, T_o and T' of Equation 22 vary over the element as follows:

$$T_o = e_1 + e_2 x + e_3 y \tag{23}$$

$$T' = e_1' + e_2' x + e_3' y \tag{24}$$

i.e.,

$$T_o = \sum_{i=1}^{3} e_i\, x^{v_i}\, y^{w_i} \tag{25}$$

$$T' = \sum_{i=1}^{3} e_i'\, x^{v_i'}\, y^{w_i'} \tag{26}$$

The constants e_1, e_2, e_3 and e_1', e_2', and e_3' can be evaluated from the user supplied values of the mean temperature and temperature gradient at the vertices of the element; however, as stated earlier, only the capability of specifying a temperature gradient for the element is currently available and hence e_1' will be equal to the element temperature gradient and e_2' and e_3' will be equal to zero.

Substituting Equations 11 through 16 into Equation 21 and substituting for $\{\varepsilon\}$ and $\{\sigma_t\}$ in Equation 20, the generalized equivalent thermal load vector $\{P^t_{gen}\}$ is obtained as

$$\{P^t_{gen}\} = \frac{\partial}{\partial\{a\}} \int_V \left\{ \begin{array}{l} \sum_{i=1}^{33} (m_i\, a_i\, x^{m_i-1}\, y^{n_i} - 2h_4\, c_i\, x^{r_i}\, y^{s_i} \\ \qquad - z\, r_i\, (r_i - 1)\, c_i\, x^{r_i-2}\, y^{s_i}) \\ \\ \sum_{i=1}^{33} (q_i\, b_i\, x^{p_i}\, y^{q_i-1} - 2h_6\, c_i\, x^{r_i}\, y^{s_i} \\ \qquad - z\, s_i\, (s_i - 1)\, c_i\, x^{r_i}\, y^{s_i-2}) \\ \\ \sum_{i=1}^{33} (n_i\, a_i\, x^{m_i}\, y^{n_i-1} + p_i\, b_i\, x^{p_i-1}\, y^{q_i} \\ \qquad - 2h_5\, c_i\, x^{r_i}\, y^{s_i} - 2z\, r_i\, s_i\, c_i\, x^{r_i-1}\, y^{s_i-1}) \end{array} \right\}^t$$

$$\bullet [G_e]\, \{\alpha_e\} \left[\sum_{j=1}^{3} (e_j\, x^{v_j}\, y^{w_j} + e'_j\, x^{v'_j}\, y^{w'_j}\, z) \right] dxdydz \quad . \qquad (27)$$

Integrating over the thickness and noting that

$$\int_{-t/2}^{t/2} f(x,y)\, z\, dx\, dy\, dz = 0 \quad , \qquad (28)$$

Equation 27 reduces to

$$\{P^t_{gen}\} = \frac{\partial}{\partial\{a\}} \left(\iint \left\{ \begin{array}{l} \sum_{i=1}^{33} m_i\, a_i\, x^{m_i-1}\, y^{n_i} - 2h_4\, c_i\, x^{r_i}\, y^{s_i} \\ \\ \sum_{i=1}^{33} q_i\, b_i\, x^{p_i}\, y^{q_i-1} - 2h_6\, c_i\, x^{r_i}\, y^{s_i} \\ \\ \sum_{i=1}^{33} n_i\, q_i\, x^{m_i}\, y^{n_i-1} + p_i\, b_i\, x^{p_i-1}\, y^{q_i} - 2h_5\, c_i\, x^{r_i}\, y^{s_i} \end{array} \right\}^t \right.$$

$$\left. \bullet [G_e]\, \{\alpha_e\} \left(\sum_{j=1}^{3} e_j\, x^{v_j}\, y^{w_j} \right) \left(\sum_{k=1}^{3} d_k\, x^{t_k}\, y^{u_k} \right) dxdy \right) \qquad (29)$$

(continued)

496

$$- \frac{\partial}{\partial\{a\}} \left(\frac{1}{12} \iint \begin{Bmatrix} r_i \, (r_i - 1) \, c_i \, x^{r_i-2} \, y^{s_i} \\ s_i \, (s_i - 1) \, c_i \, x^{r_i} \, y^{s_i-2} \\ 2r_i \, s_i \, c_i \, x^{r_i-1} \, y^{s_i-1} \end{Bmatrix}^t [G_e] \, \{\alpha_e\} \right.$$

$$\cdot \left(\sum_{j=1}^{3} e'_j \, x^{v'_j} \, y^{w'_j} \right) \quad \sum_{k_1=1}^{3} \; \sum_{k_2=1}^{3} \; \sum_{k_3=1}^{3} \; d_{k_1} \, d_{k_2} \, d_{k_3} \, x^{t_{k_1}+t_{k_2}+t_{k_3}}$$

$$\left. \cdot \; y^{u_{k_1}+u_{k_2}+u_{k_3}} \, dxdy \right) \quad . \qquad (29)$$

(concluded)

The generalized equivalent thermal load vector will be obtained by performing the differentiation and integration operations of Equation 29 and the final expression for $\{P^t_{gen}\}$ will be similar to those obtained for the TRIM6 and TRPLT1 elements, except that an additional expression involving the curvatures of the shell surface h_4, h_5, and h_6 will be added now. The generalized thermal load vector $\{P^t_{gen}\}$ can then be transformed to the element and global coordinate systems by the usual procedures.

THE TRIANGULAR THIN SHALLOW SHELL ELEMENT

REFERENCES

1. Narayanaswami, R., "New Triangular Shallow Shell Finite Element," Technical Report 74-T7, School of Engineering, Old Dominion University, Norfolk, Virginia, November 1974.

2. Novozhilov, V. V., Thin Shell Theory, 2nd edition, Noordhoff, 1964.

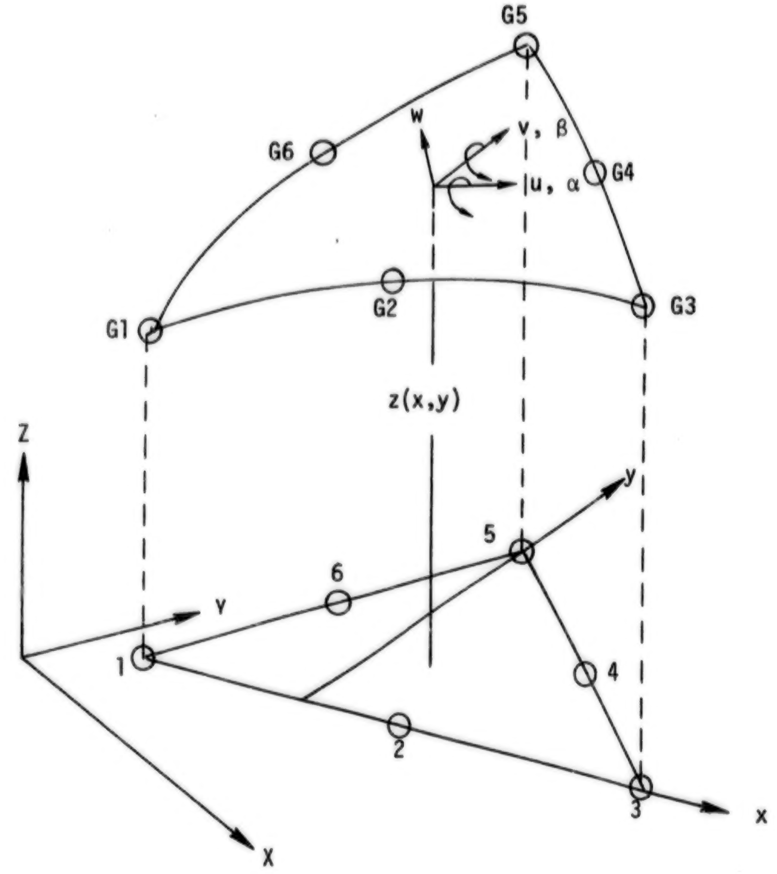

G5
G6
w
v, β
u, α
G4
G2
G1
G3
z(x,y)
Z
y
5
6
Y
1
4
2
3
x
X

Figure 1. Triangular thin shallow shell element geometry.

7. DIFFERENTIAL STIFFNESS AND BUCKLING

7.1 INTRODUCTION

The term "differential stiffness" applies to linear terms in the equations of motion of an elastic body that arise from a simultaneous consideration of large, nonlinear motions and the applied loads. The theory of differential stiffness is not an exact theory and it involves inherent assumptions that are arbitrary and that may be changed depending on their practical effect. The approach presented here to the theory of differential stiffness is based on Lagrange's equations for the motion of a system with a finite number of degrees of freedom. This approach is useful because it points out some of the assumptions and limitations of the general theory.

Consider a system with a finite number of degrees of freedom, q_r; with a set of springs whose potential energy is V; and with a set of loads, P_a, applied to displacements u_a. The equations of motion for the system may be written

$$\frac{\partial V}{\partial q_r} = Q_r \qquad r = 1, 2, 3, \ldots n \quad , \tag{1}$$

where the generalized force

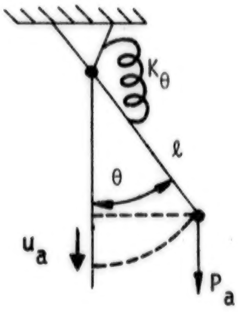
$$Q_r = \frac{\partial W}{\partial q_r} = \sum_a \frac{\partial u_a}{\partial q_r} P_a \quad . \tag{2}$$

W is the work done by the external forces. It is assumed in the theory of differential stiffness that the potential energy is a quadratic function of the degrees of freedom, i.e.,

$$V = \frac{1}{2} \sum_{i,j} a_{ij} q_i q_j \quad , \tag{3}$$

but that the partial derivatives, $\frac{\partial u_a}{\partial q_r}$, are not necessarily constants.

As a simple example to illustrate concepts, consider a pendulum with spring restraint shown on the following page.



The potential energy of the spring is

$$V = \frac{1}{2} K_\theta \theta^2 \quad . \tag{4}$$

The displacement at the point of application and in the direction of the applied load is

$$u_a = \ell(\cos\theta - 1) \quad , \tag{5}$$

so that as a result of Equations 1 and 2,

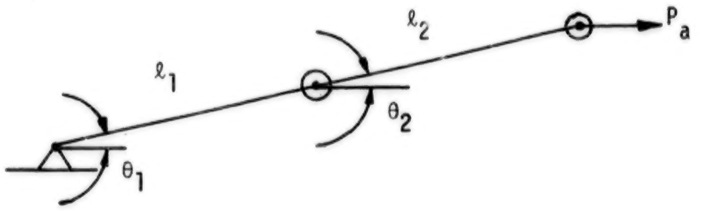
$$K_\theta \theta = - P_a \ell \sin\theta \quad . \tag{6}$$

At this point the term on the right is linearized with the result

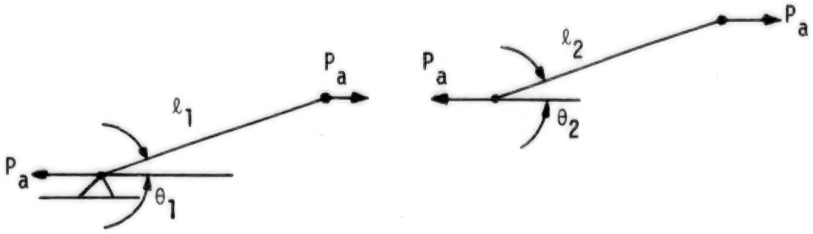
$$(K_\theta + P_a \ell)\theta = 0 \quad . \tag{7}$$

$P_a \ell$ is the "differential stiffness".

In a practical problem with many degrees of freedom it is difficult to calculate the partial derivatives appearing in Equation 2, primarily because the points of application of the loads may be remote from the degrees of freedom. The problem is simplified by replacing the applied loads by a statically equivalent set of loads acting directly on structural elements, such as are used in free body diagrams. The generalized forces Q_r are then computed from the work done by the load subsets for individual elements and Equation 2. As an example consider the following pendulum.



The single load P_a is replaced by the following pair of equivalent loading systems



The work done by loading system (1) during general motion is:

$$W_1 = P_a \ell_1 \cos\theta_1 \quad , \tag{8}$$

while the work done by loading system (2) is:

$$W_2 = P_a \ell_2 \cos\theta_2 \quad . \tag{9}$$

It may be concluded, by referring to Equation 2 and linearizing, that $P_a \ell_1$ is the differential stiffness for θ_1 and that $P_a \ell_2$ is the differential stiffness for θ_2.

It is important to observe that the equivalent loading systems remain fixed in magnitude and direction, and move with their points of application during motion of the system, so that the equivalent loading systems are in equilibrium with each other during the motions. This implies that the actual applied loads also remain fixed in magnitude and direction and move with their points of application.

The Lagrangian discrete element approach can be applied to a general elastic body, if it be imagined that the body is made up of infinitesimal cubes, each of which is joined to its six neighbors by a universal joint at the midpoint of each face. For a given static loading on the body the stress distribution is computed throughout the body, ignoring differential stiffness effects in the process. This internal stress distribution is taken as the equivalent loading, and is applied to each cube in turn to determine the differential stiffness for the cube.

The degrees of freedom for each cube are taken to be its three translations, three rotations and six elastic strains. It is clear that not all of the degrees of freedom can be independent, in view of the attachments to other cubes. It is, however, permissible to consider them to be independent in computing the work done on each cube.

The work done by the static loads is computed for general motions of the degrees of freedom using Equation 2. The terms in the differential stiffness matrix for the cube are then computed from

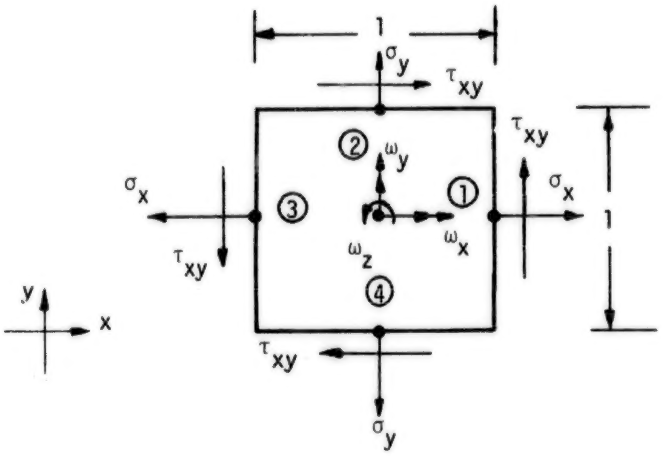
$$K_{rs} = -\frac{\partial Q_r}{\partial q_s} = -\frac{\partial^2 W}{\partial q_s \partial q_r} , \qquad (10)$$

for the condition $q_r = q_s = 0$, which is the linearizing assumption.

Consider a view of a unit cube from the positive z axis shown on the following page. During general motion the loads applied to the cube remain fixed in magnitude and direction and remain attached to the midpoints of the faces. It is clear that no work is done on the cube during translation because the forces acting on the cube are in equilibrium. The strains are eliminated from consideration on the assumption that they are small compared to the rotations. This assumption is not essential, and it may be removed in cases, such as occur in built-up structures, where the elastic resistance to some strain components is small. The remaining degrees of freedom, the rotations, are the only significant ones.

During a rotation about the z axis, point (1) in the following figure moves to the left by the amount $\frac{1}{2}(1 - \cos\omega_z)$ and moves upward by an amount of $\frac{1}{2}\sin\omega_z$. The work done by the forces acting at point (1) during rotation about the z axis is

$$W_{1,z} = -\frac{1}{2}(1 - \cos\omega_z)\sigma_x + \frac{1}{2}\sin\omega_z \cdot \tau_{xy} \quad . \tag{11}$$



The total work done by all forces is

$$W_z = -(1 - \cos\omega_z)(\sigma_x + \sigma_y) \quad , \tag{12}$$

so that the differential stiffness, for rotation about the ω_z axis is, from Equation 10

$$K_{zz} = \sigma_x + \sigma_y \quad . \tag{13}$$

For the case of general motions ω_x, ω_y, and ω_z considered simultaneously, first compute the work done by the force components in the z plane shown in the above figure. The work for other force components can then be evaluated by permutation of indices. The work done in rotations ω_x, ω_y, and ω_z by the normal stresses, σ_x and σ_y, is, (for small ω)

$$W_\sigma = -\frac{1}{2}[\omega_z^2(\sigma_x + \sigma_y) + \omega_x^2\sigma_y + \omega_y^2\sigma_x] \quad . \tag{14}$$

The cross-product $\omega_x\omega_y$ also produces work via the shear τ_{xy}. Consider that the rotation ω_x occurs first, causing outward motion at point (2) and inward motion at point (4) both approximately equal to $\frac{1}{2}\omega_x$. A subsequent rotation ω_y, about the original y axis, produces motion at points (2) and

(4) approximately equal to $\frac{1}{2}\omega_x\omega_y$ in the direction of the applied shear loads. The work done is therefore

$$W_\tau = \omega_x\omega_y\tau_{xy} \quad . \tag{15}$$

The same result is obtained if the order of rotations is reversed, the work being done, however, by the shear forces on the vertical faces. Thus we conclude that Equation 15 is correct for any general small motion.

The total work done by all components of force on a cube of volume ΔV is, by extension of the above results,

$$\Delta W = -\frac{\Delta V}{2}\Big[\omega_x{}^2(\sigma_y + \sigma_z) + \omega_y{}^2(\sigma_z + \sigma_x) + \omega_z{}^2(\sigma_x + \sigma_y) - 2\omega_x\omega_y\tau_{xy} - 2\omega_y\omega_z\tau_{yz} - 2\omega_z\omega_x\tau_{zx}\Big] \quad . \tag{16}$$

The matrix of differential stiffness coefficients for a cube of volume ΔV is, from Equation 10

$$\Delta[K_{\omega\omega}{}^d] = \Delta V \begin{array}{c} \begin{array}{ccc} \omega_x & \omega_y & \omega_z \end{array} \\ \left[\begin{array}{c|c|c} \sigma_y + \sigma_z & -\tau_{xy} & -\tau_{zx} \\ \hline -\tau_{xy} & \sigma_z + \sigma_x & -\tau_{yz} \\ \hline -\tau_{zx} & -\tau_{yz} & \sigma_x + \sigma_y \end{array}\right] \end{array} \begin{array}{c} \omega_x \\ \omega_y \\ \omega_z \end{array} \quad . \tag{17}$$

The above general result is applied, in subsequent sections, to the evaluation of differential stiffness matrices for specific structural elements. The steps for including differential stiffness in a problem are

1. Solve the linear static response problem for the structure in the absence of differencial stiffness, and compute the internal forces in elements.
2. Using the results of Step 1, calculate the differential stiffness matrices for individual elements, and apply the standard reduction procedures (constraints and partitioning) to form the differential stiffness matrix $[K_{aa}{}^d]$ in final form.

505

3. In buckling problems, find eigenvalues and eigenvectors for

$$[K_{aa} + \lambda\, K_{aa}{}^{d}]\{u\} = 0 \quad . \tag{18}$$

The eigenvalues are the factors by which the applied static loading is multiplied to produce buckling.

4. In response problems in which the stiffening effect of a static preload is desired, add $\left[K_{aa}{}^{d}\right]$ to $\left[K_{aa}\right]$ and proceed in the normal manner.

At present, rigid formats which include differential stiffness are provided only for the buckling problem and for static problems in which the applied load is a multiple of that used to calculate $[K^d]$, (see Section 3.2). Other applications of differential stiffness, such as its inclusion in dynamic problems, can be treated by using the ALTER feature, Section 1.3, to modify one of the rigid formats.

An important limitation of the automatic procedures provided with NASTRAN is in the assumption that the applied loads from which the differential stiffness is derived remain fixed in magnitude and direction during motion of the structure, and that their points of application move with the structure. An example in which the direction of the load changes is the buckling of a container loaded by external fluid pressure. An example of a stability problem in which both the magnitude and direction of the loads change is the development of a bulge on a balloon. In such examples, the burden is placed on the user of NASTRAN to prescribe the additional stiffness terms (via direct stiffness matrix input) that result from changes in the magnitudes or directions of applied loads. Such terms are usually unsymmetrical. Dynamic routines (complex eigenvalue extraction, and frequency response) are then used to solve the problems. The frequency response module provides the means for solving static response problems with unsymmetrical stiffness matrices. The user selects the frequency to be zero.

The algorithm has been changed in Rigid Format 4 (Static Analysis with Differential Stiffness) to iterate the differential stiffness calculation. The idea is that, in general, since $[K^d]$ is a function of $\{u\}$, a better solution will be obtained by using the most recently updated value of $\{u\}$ in computing $[K^d]$. In its simplest form the iteration algorithm is

$$[K + K^{d}(u_{i})]\{u_{i+1}\} = \{P\} \quad , \tag{19}$$

where the subscripts i and $i+1$ refer to the ith and i+1st iterations. It is by no means self-evident that the algorithm will converge to a better solution than simply basing $[K^d]$ on the elastic solution $\{u_1\}$, or, indeed, that it will converge at all. Mathematical analysis shows that iteration is equivalent to retaining some, but not all, second order terms in the calculation of the differential stiffness.

Figure 1 shows the results of applying various methods to a simple problem. The structure consists of a hinged rigid rod with an over-center spring attached to one end. The spring is assumed to be long enough that movement of the attachment point does not significantly change its direction. It is clear, from the results, that the iterated differential stiffness solution is a significant improvement over the one-step differential stiffness solution, but that it still differs considerably from the correct solution.

The mathematical development of the iterated solution for this simple case is as follows. Let the force in the spring be F, and let the lateral deflection at the end of the rod be u. Then the compressive load in the rod is $F/\sqrt{2}$ (for small motions) and the linearized differential stiffness presented to u is, according to Equation 7,

$$K^d = -F/\ell\sqrt{2} \tag{20}$$

The force in the spring is related to u by

$$F = Ku/\sqrt{2} \quad , \tag{21}$$

so that

$$K^d = -Ku/2\ell \quad . \tag{22}$$

In addition, the linear elastic stiffness presented to u by the spring is

$$K^\ell = K/2 \quad . \tag{23}$$

The explicit form of the iteration algorithm corresponding to Equation 19 is

$$\left(\frac{K}{2} - \frac{Ku_i}{2\ell}\right)u_{i+1} = P \quad , \tag{24}$$

or

$$\frac{u_{i+1}}{\ell} = \frac{2P/\ell K}{1 - \frac{u_i}{\ell}} \quad . \tag{25}$$

Both the lower (stable)branch and the upper (unstable) branch obtained by replacing u_i and u_{i+1} by u are plotted in Figure 1. Convergence to within one percent of the lower branch is obtained in three iterations for $P/\ell K = .10$. The algorithm is divergent for $P/\ell K > .125$, the stability limit.

As a matter of practical computation, Equation 19 is inconvenient because it requires decomposition of $[K + K^d(u_i)]$ at each iteration. Improved efficiency could be achieved by moving $[K^d(u_i)]$ to the right hand side. Convergence would, however, be adversely affected. As a compromise $[K^d(u_i)]$ is replaced on the left by an approximation $[K^d(u_e)]$ where $\{u_e\}$ is initially the linear elastic solution. The quantity, $\{u_e\}$, may be updated at a later iteration if it is advantageous to do so. The iteration algorithm used by NASTRAN is

$$\begin{aligned}[K + K^d(u_e)]\{u_{i+1}\} &= \{P\} + [K^d(u_e) - K^d(u_i)]\{u_i\} \\ &= \{P\} + [K^d(u_e - u_i)]\{u_i\} = \{P_i\}\end{aligned} \quad , \tag{26}$$

where the second form follows from the linear dependence of $[K^d]$ on its argument.

Criteria are needed to determine when the solution is sufficiently converged, and when (if at all) $\{u_e\}$ should be updated to be the current estimated solution, $\{u_i\}$. These criteria are computed automatically from parameters supplied by the user.

Convergence is deemed to have been achieved when the difference between successive right hand sides of Equation 26 is sufficiently small. A weighted criterion for this event is that

$$\varepsilon_i = \left| \frac{\{u^{i+1}\}^T \{P_{i+1} - P_i\}}{\{u^{i+1}\}^T \{P_i\}} \right| < \varepsilon_0 \quad , \tag{27}$$

where ε_0 is a user-supplied parameter and $\{P_i\}$ is the right hand side of Equation 26 at the ith iteration.

If an exponential decay in the error is assumed, and if $\lambda = \varepsilon_{i-1}/\varepsilon_i$ is the ratio of the error on two successive iterations, then the number of additional iterations required to achieve the desired accuracy is approximately

$$N_f = \frac{\log \frac{\varepsilon_i}{\varepsilon_o}}{\log \lambda} \quad . \tag{28}$$

Exponential decay will be approached after a few iterations for algorithms having the general form of Equation 26, as may be inferred from the analysis of the NASTRAN algorithm for nonlinear steady-state heat transfer in Section 8.

The number of additional iterations, N_f, may be used in conjunction with the known computer times for matrix decomposition and for one cycle of iteration, to decide whether it is more efficient to continue iterating or to "shift," i.e., to replace $\{u_e\}$ by the current approximate solution. A flow diagram for the logical steps involved in deciding when to stop, and whether to "shift," is shown in Figures 2a and 2b. In the diagram, β is a user-supplied parameter equal to his estimate of the number of iterations that will be required for convergence after a shift has occured.

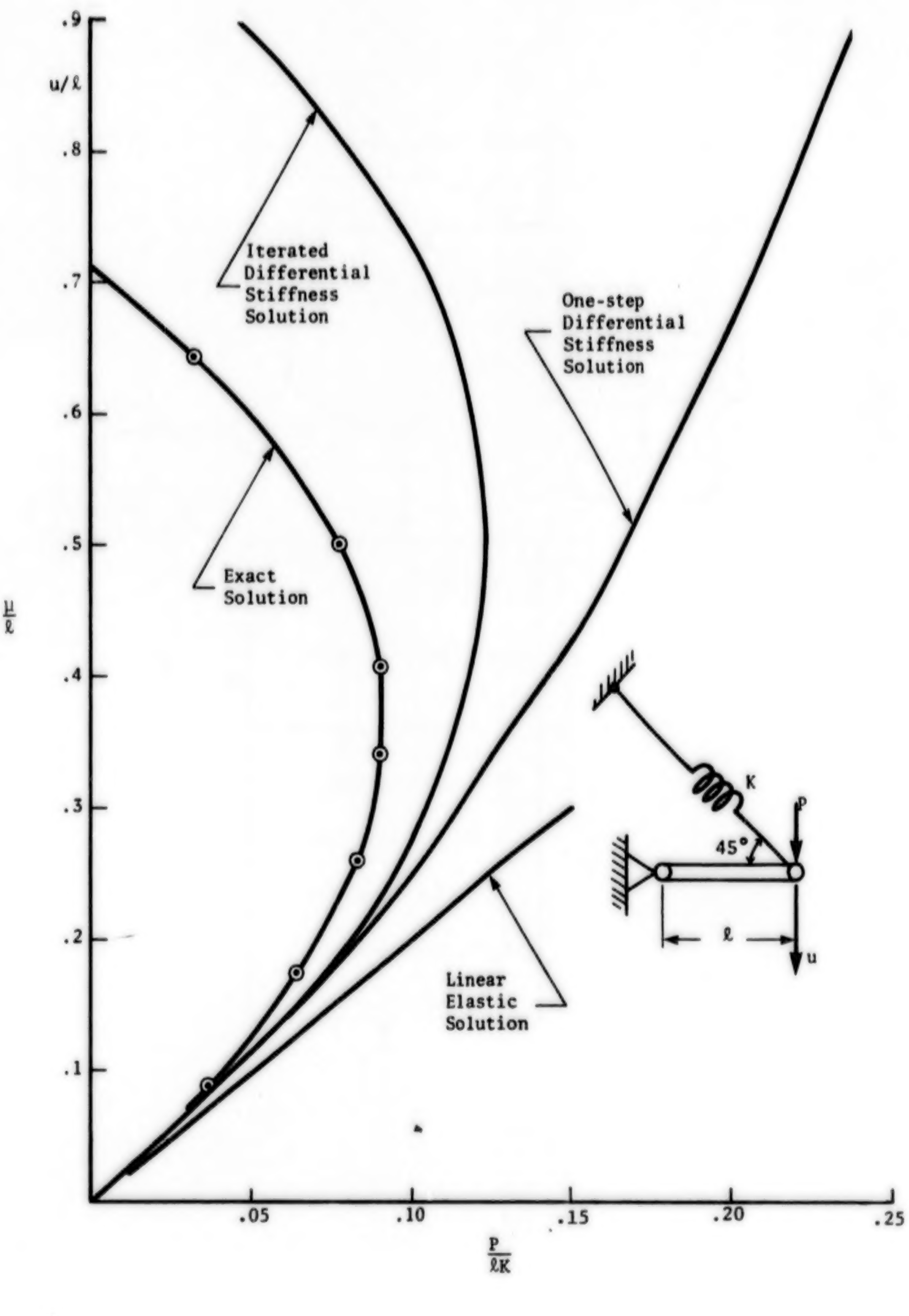


Figure 1. Solutions to simple nonlinear problem.

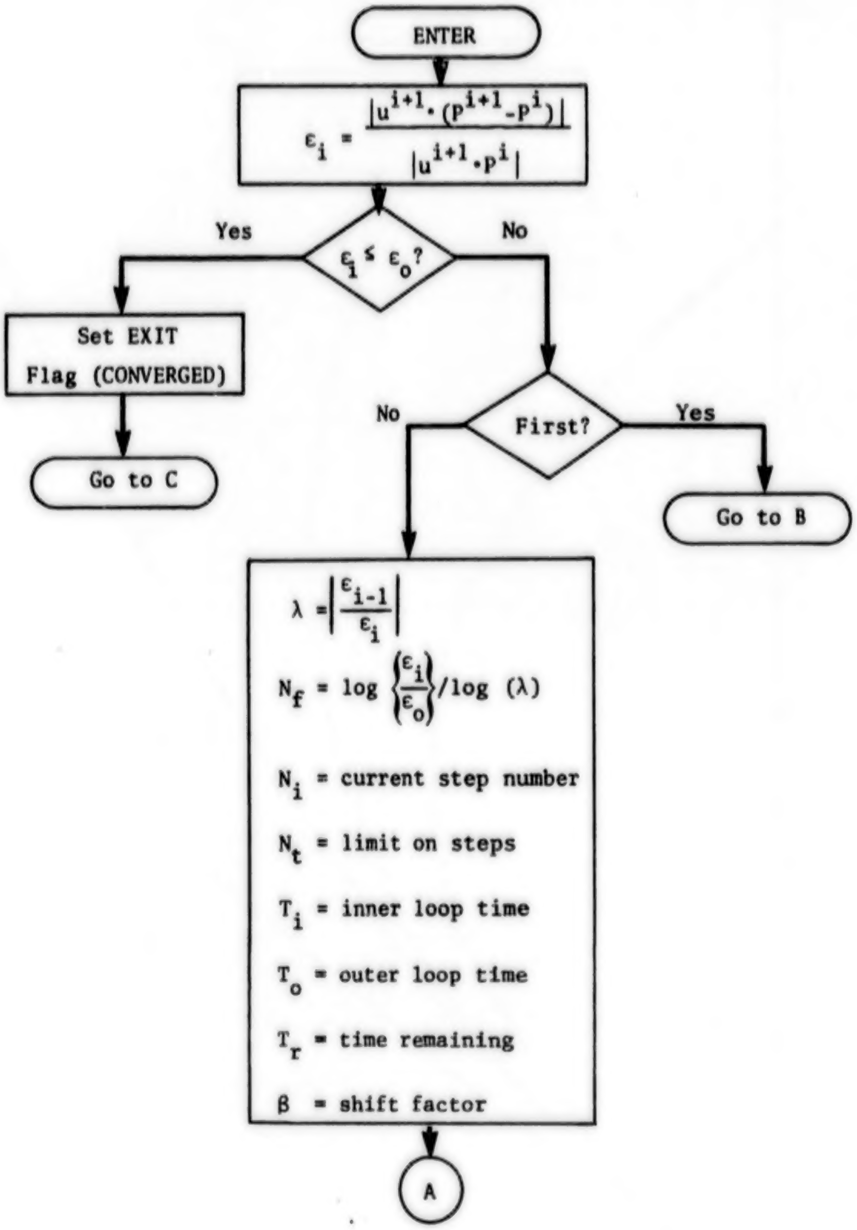

ENTER
$\varepsilon_i = \dfrac{|u^{i+1} \cdot (P^{i+1} - P^i)|}{|u^{i+1} \cdot P^i|}$
Yes
$\varepsilon_i \le \varepsilon_o$?
No
Set EXIT Flag (CONVERGED)
Go to C
No
First?
Yes
Go to B
$\lambda = \left|\dfrac{\varepsilon_{i-1}}{\varepsilon_i}\right|$
$N_f = \log \left\{\dfrac{\varepsilon_i}{\varepsilon_o}\right\} / \log (\lambda)$
N_i = current step number
N_t = limit on steps
T_i = inner loop time
T_o = outer loop time
T_r = time remaining
β = shift factor
A

Figure 2a. Flow diagram for module DSCHK (first part).

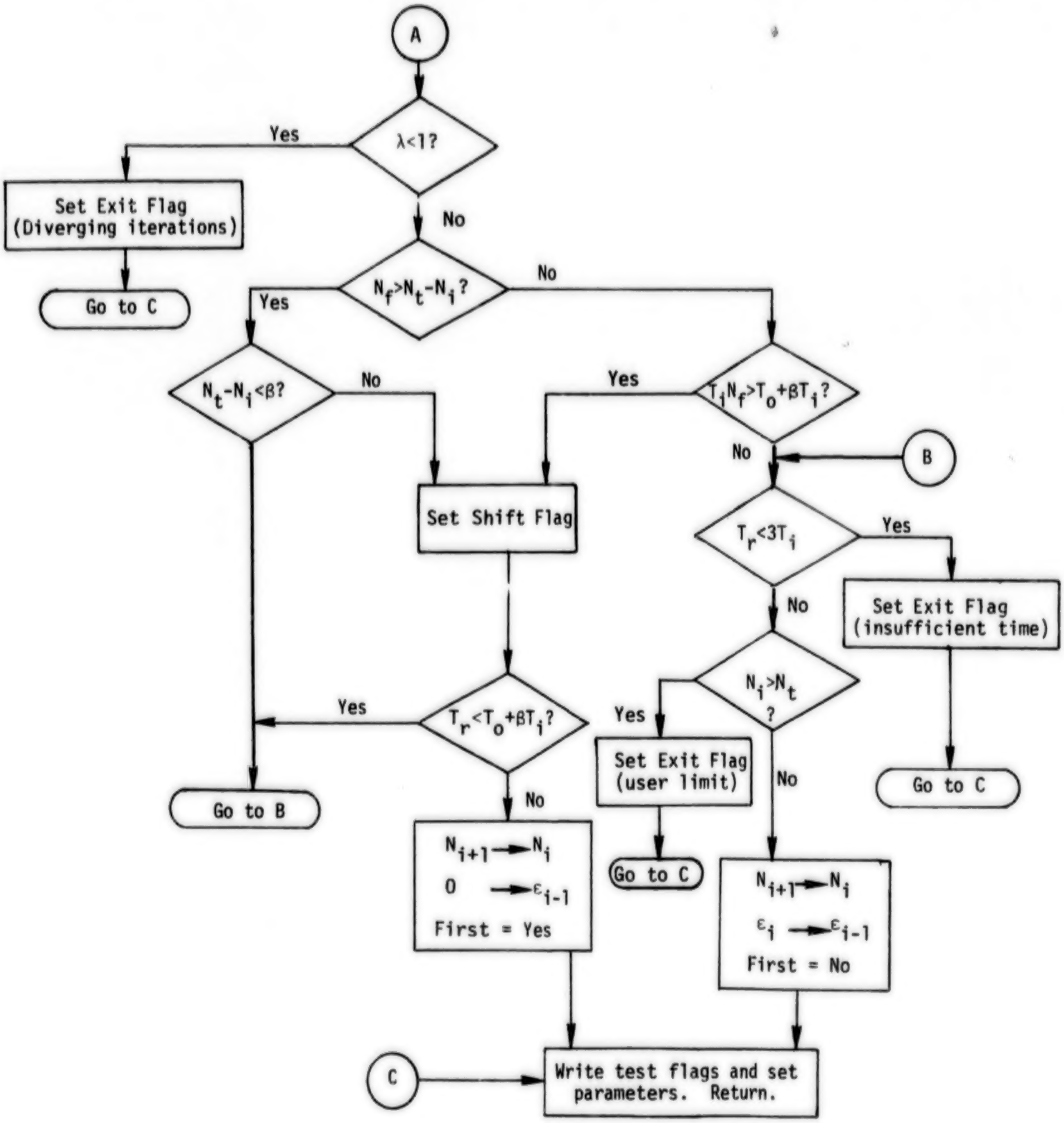

A
λ<1?
Yes
Set Exit Flag
(Diverging iterations)
Go to C
No
$N_f > N_t - N_i$?
Yes
No
$N_t - N_i < \beta$?
No
$T_i N_f > T_o + \beta T_i$?
Yes
No
B
Set Shift Flag
$T_r < 3T_i$
Yes
Set Exit Flag
(insufficient time)
No
$N_i > N_t$?
Yes
Set Exit Flag
(user limit)
Go to C
No
$T_r < T_o + \beta T_i$?
Yes
Go to B
No
$N_{i+1} \rightarrow N_i$
$0 \rightarrow \varepsilon_{i-1}$
First = Yes
Go to C
$N_{i+1} \rightarrow N_i$
$\varepsilon_i \rightarrow \varepsilon_{i-1}$
First = No
C
Write test flags and set
parameters. Return.

Figure 2b. Flow diagram for module DSCHK (concluded).

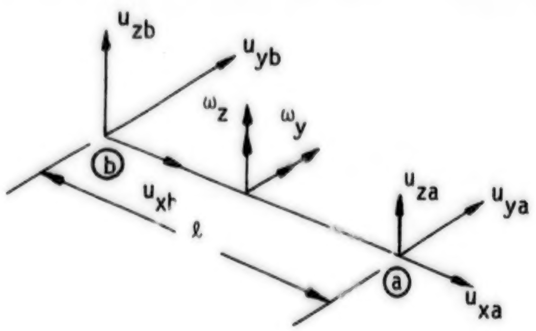
7.2 RODS, BEAMS, AND SHEAR PANELS

7.2.1 Extensional Rods

The calculation of differential stiffness for an extensional rod is particularly simple. If the x axis is selected to coincide with the axis of the rod, σ_x is the only nonzero stress and the work done by static preload is

$$W_r = -\frac{\ell A \sigma_x}{2}(\omega_y^2 + \omega_z^2) = -\frac{\ell F_x}{2}(\omega_y^2 + \omega_z^2) \quad , \tag{1}$$

where ℓ is the length of the rod, A is its cross-sectional area, and ω_y and ω_z are rotations about transverse axes. F_x is the axial force in the rod. Consider the following diagram.



The rotations are computed from deflections at the ends

$$\omega_y = (u_{zb} - u_{za})/\ell \quad , \tag{2}$$

$$\omega_z = (u_{ya} - u_{yb})/\ell \quad . \tag{3}$$

Thus the work done by σ_x is

$$W_r = -\frac{F_x}{2\ell}\left[(u_{zb} - u_{za})^2 + (u_{ya} - u_{yb})^2\right] \quad . \tag{4}$$

PRECEDING PAGE BLANK NOT FILMED

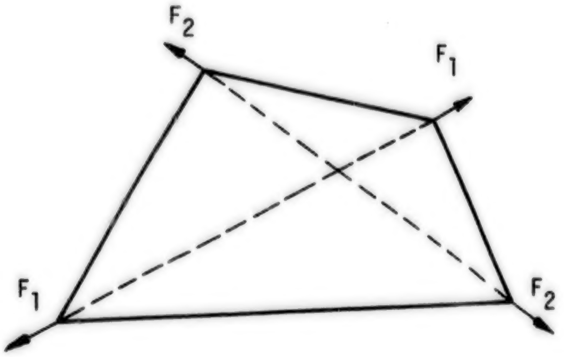
The differential stiffness matrix is given by

$$\begin{Bmatrix} f_{ya} \\ f_{yb} \\ f_{za} \\ f_{zb} \end{Bmatrix} = \frac{F_x}{\ell} \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_{ya} \\ u_{yb} \\ u_{za} \\ u_{zb} \end{Bmatrix} \quad . \tag{5}$$

Note that F_x is positive for tension.

7.2.2 Quadrilateral Shear Panels

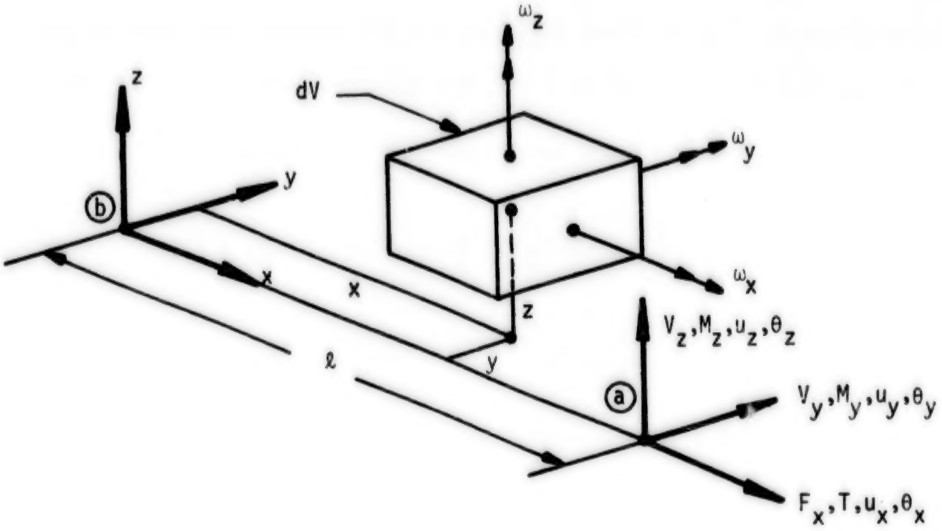
Treatment of differential stiffness for the quadrilateral shear panel is simple because the forces exerted on the shear panel are directed along the principal diagonals as shown below.



Forces F_1 and F_2 are evaluated from corner displacements during stress data recovery (see Section 5.3). The diagonals are treated as independent extensional rods, for the purpose of calculating differential stiffness. The resulting matrices are then transformed from element coordinates, which are oriented parallel and perpendicular to the diagonals, to grid point coordinates. Details of the procedure are described in Section 8 of the Programmer's Manual.

7.2.3 Prismatic Beams

The differential stiffness matrix for a prismatic beam (BAR), is relatively complicated if the effects of moments and shears as well as of axial force are considered. This is done in NASTRAN in order to achieve generality of application. Consider a small length of prismatic beam shown below.



All forces, moments, displacements and rotations follow right-hand rules. The rotations of an infinitesimal element located within the beam are also indicated in the figure.

For a prismatic beam the stress components $\sigma_y = \sigma_z = \tau_{yz} = 0$. Thus the energy due to static preload can, from the general theory developed in Section 7.1, be expressed as

$$W = -\frac{1}{2}\int[\sigma_x(\omega_y{}^2 + \omega_z{}^2) - 2\omega_x\omega_y\tau_{xy} - 2\omega_x\omega_z\tau_{zx}]dV \quad , \qquad (6)$$

where the integration is carried out over the entire volume.

The rotations appearing in Equation 6 are related to the motions that define the position and orientation of the beam axis by the following formulas.

516

$$\omega_x = \theta_x \quad , \tag{7}$$

$$\omega_y = -\frac{\partial u_z}{\partial x} - y\frac{\partial \theta_x}{\partial x} \equiv -u_z{}' - y\theta_x{}' \quad , \tag{8}$$

$$\omega_z = \frac{\partial u_y}{\partial x} - z\frac{\partial \theta_x}{\partial x} \equiv u_y{}' - z\theta_x{}' \quad . \tag{9}$$

The rotations ω_y and ω_z appearing in Equations 8 and 9 are the rotations of filaments originally parallel to the x axis. They differ from the average rigid body rotations by one half of the corresponding shear angles. It is clear that they are the correct rotations to use for the axial stress terms in Equation 6, and it can be shown that they are also correct for the shear terms.

Equation 6, written in terms of the local slopes of the beam axis, is

$$W = -\frac{1}{2}\int\Big[\sigma_x\Big\{(u_z{}' + y\theta_x{}')^2 + (u_y{}' - z\theta_x{}')^2\Big\} + 2\theta_x\Big\{\tau_{xy}(u_z{}' + y\theta_x{}') - \tau_{zx}(u_y{}' - z\theta_x{}')\Big\}\Big]dV \quad . \tag{10}$$

The terms in the differential stiffness matrix are obtained from

$$K_{ij}{}^d = -\frac{\partial^2 W}{\partial \bar{u}_i \partial \bar{u}_j} \quad , \tag{11}$$

where $\bar{u}_i$ is an element of $\{\bar{u}_i\} = \lfloor u_y{}', u_z{}', \theta_x{}', \theta_x \rfloor^T$. Thus, for a differential element of length dx

$$K_{11}{}^d = \int \sigma_x dV = F_x dx \quad , \tag{12}$$

$$K_{12}{}^d = 0 \quad , \tag{13}$$

$$K_{13}{}^d = -\int z\sigma_x dV = -M_y dx \quad , \tag{14}$$

$$K_{14}{}^d = -\int \tau_{zx} dV = -V_z dx \quad , \tag{15}$$

$$K_{22}{}^{d} = \int \sigma_x dV = F_x dx \quad , \tag{16}$$

$$K_{23}{}^{d} = \int \sigma_x y dV = -M_z dx \quad , \tag{17}$$

$$K_{24}{}^{d} = \int \tau_{xy} dV = V_y dx \quad , \tag{18}$$

$$K_{33}{}^{d} = \int \sigma_x (y^2 + z^2) dV = \left(\frac{I}{A}\right) F_x dx \quad , \tag{19}$$

$$K_{34}{}^{d} = \int (y\tau_{xy} + z\tau_{zx}) dV = 0 \quad , \tag{20}$$

$$K_{44}{}^{d} = 0 \quad . \tag{21}$$

In Equation 19, I is the polar area moment of inertia about the centroid and A is the cross-sectional area. It is assumed, in Equations 19 and 20, that the cross section is symmetrical about the y and z axes. The assembled differential stiffness matrix for an incremental beam of unit length is

$$[K_{ij}{}^{d}] = \begin{array}{c} \begin{array}{cccc} u_y' & u_z' & \theta_x' & \theta_x \end{array} \\ \left[\begin{array}{c:c:c:c} F_x & 0 & -M_y & -V_z \\ \hdashline 0 & F_x & -M_z & V_y \\ \hdashline -M_y & -M_z & \frac{IF_x}{A} & 0 \\ \hdashline -V_z & V_y & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} \\ u_y' \\ u_z' \\ \theta_x' \\ \theta_x \end{array} \quad . \tag{22}$$

The energy due to static preload for the entire beam can now be written as

$$W = \frac{1}{2} \int_a^b \{\bar{u}_i\}^T \, [K_{ij}{}^{d}] \{\bar{u}_i\} dx \quad . \tag{23}$$

In order to obtain the differential stiffness matrix in terms of displacements at the ends of the bar, $\{u_e\}$, let the relationship between local displacements $\{\bar{u}_i\}$ and $\{u_e\}$ be

518

ORIGINAL PAGE IS OF POOR QUALITY

$$\{\bar{u}_i\} = [D]\{u_e\} \quad . \tag{24}$$

The differential stiffness matrix referred to $\{u_e\}$ is related to the energy by

$$W = \frac{1}{2}\{u_e\}^T [K_{ee}{}^d]\{u_e\} \quad , \tag{25}$$

so that, substituting Equation 24 into Equation 23 and comparing with Equation 25

$$[K_{ee}{}^d] = \int_a^b [D]^T [K_{ij}{}^d][D]dx \quad . \tag{26}$$

The result clearly depends on the form assumed for the terms of the [D] matrix. Let us consider, for example, only the first row of [D], which gives the bending slope $u_y{}'$. If it is assumed that the variation in displacement is linear, i.e., if

$$u_y{}' = \frac{1}{\ell}(u_{ya} - u_{yb}) \quad , \tag{27}$$

then the terms in $[K_{ee}{}^d]$ corresponding to u_{ya} and u_{yb} are

$$[K_{ee}{}^d] = \frac{F_x}{\ell}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad , \tag{28}$$

which is identical to the result for an extensional rod expressed in Equation 5.

More accurate results are obtained if the elements of [D] are chosen to correspond to the deflection functions for beam bending (Reference 1).

Since the deflection function for a uniform beam is a cubic function of distance from one end, the slope will be a quadratic function. The exact relationship between $u_y{}'$ and end motions is, neglecting transverse shear strain,

$$u_y' = \frac{6}{\ell}(\bar{x} - \bar{x}^2)(u_{ya} - u_{yb}) - \theta_{za}(2\bar{x} - 3\bar{x}^2) + \theta_{zb}(1 - 4\bar{x} + 3\bar{x}^2) \quad , \qquad (29)$$

where $\bar{x} = x/\ell$.

Similarly

$$u_z' = \frac{6}{\ell}(\bar{x} - \bar{x}^2)(u_{za} - u_{zb}) + \theta_{ya}(2\bar{x} - 3\bar{x}^2) - \theta_{yb}(1 - 4\bar{x} + 3\bar{x}^2) \quad , \qquad (30)$$

and

$$\theta_x = \bar{x}\,\theta_{xa} + (1 - \bar{x})\,\theta_{xb} \quad , \qquad (31)$$

$$\theta_x' = \frac{1}{\ell}(\theta_{xa} - \theta_{xb}) \quad . \qquad (32)$$

The coefficients appearing in Equations 29 through 32 are the nonzero terms in the [D] matrix. Substitution of the coefficients and Equation 22 into Equation 26 results, upon integration, in the differential stiffness matrix shown on the following page. For example, the leading diagonal term is

$$K_{11}^d = \frac{1}{\ell}\int_0^1 36\,F_x\,(\bar{x} - \bar{x}^2)^2\,d\bar{x} = \frac{6F_x}{5\ell} \quad . \qquad (33)$$

In performing the integration the variations of the bending moments with distance,

$$M_y = \bar{x}\,M_{ya} - (1 - \bar{x})\,M_{yb} \quad , \qquad (34)$$

$$M_z = \bar{x}\,M_{za} - (1 - \bar{x})\,M_{zb} \quad , \qquad (35)$$

are used. Note that the convention for positive direction of end moment gives $M_y = -M_{yb}$ and $M_z = -M_{zb}$ at $\bar{x} = 0$.

The final operations on the differential stiffness matrix are to remove the effects of pin flags (which constrain one or more of the end forces and moments to zero value) and to transfer the resulting matrix from element coordinates to grid point coordinates. See Section 4.87 of the Programmer's Manual for details of the procedure.

$$
\begin{Bmatrix} f_{ya} \\ f_{za} \\ m_{xa} \\ m_{ya} \\ m_{za} \\ f_{yb} \\ f_{zb} \\ m_{xb} \\ m_{yb} \\ m_{zb} \end{Bmatrix}
=
\left[\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
\frac{6F_x}{5\ell} & 0 & -\frac{M_{ya}}{\ell} & 0 & -\frac{F_x}{10} & -\frac{6F_x}{5\ell} & 0 & -\frac{M_{yb}}{\ell} & 0 & -\frac{F_x}{10} \\ \hline
0 & \frac{6F_x}{5\ell} & -\frac{M_{za}}{\ell} & \frac{F_x}{10} & 0 & 0 & -\frac{6F_x}{5\ell} & -\frac{M_{zb}}{\ell} & \frac{F_x}{10} & 0 \\ \hline
-\frac{M_{ya}}{\ell} & -\frac{M_{za}}{\ell} & \frac{IF_x}{\ell A} & -\frac{\ell V_y}{6} & -\frac{\ell V_z}{6} & \frac{M_{ya}}{\ell} & \frac{M_{za}}{\ell} & -\frac{IF_x}{\ell A} & \frac{\ell V_y}{6} & \frac{\ell V_z}{6} \\ \hline
0 & \frac{F_x}{10} & -\frac{\ell V_y}{6} & \frac{2\ell}{15} F_x & 0 & 0 & -\frac{F_x}{10} & \frac{\ell V_y}{6} & -\frac{\ell F_x}{30} & 0 \\ \hline
-\frac{F_x}{10} & 0 & -\frac{\ell V_z}{6} & 0 & \frac{2\ell}{15} F_x & \frac{F_x}{10} & 0 & \frac{\ell V_z}{6} & 0 & -\frac{\ell F_x}{30} \\ \hline
-\frac{6F_x}{5\ell} & 0 & \frac{M_{ya}}{\ell} & 0 & \frac{F_x}{10} & \frac{6F_x}{5\ell} & 0 & \frac{M_{yb}}{\ell} & 0 & \frac{F_x}{10} \\ \hline
0 & -\frac{6F_x}{5\ell} & \frac{M_{za}}{\ell} & -\frac{F_x}{10} & 0 & 0 & \frac{6F_x}{5\ell} & \frac{M_{zb}}{\ell} & -\frac{F_x}{10} & 0 \\ \hline
-\frac{M_{yb}}{\ell} & -\frac{M_{zb}}{\ell} & -\frac{IF_x}{\ell A} & \frac{\ell V_y}{6} & \frac{\ell V_z}{6} & -\frac{M_{yb}}{\ell} & \frac{M_{zb}}{\ell} & \frac{IF_x}{\ell A} & -\frac{\ell V_y}{6} & -\frac{\ell V_z}{6} \\ \hline
0 & \frac{F_x}{10} & \frac{\ell V_y}{6} & -\frac{\ell F_x}{30} & 0 & 0 & -\frac{F_x}{10} & -\frac{\ell V_y}{6} & \frac{2\ell F_x}{15} & 0 \\ \hline
-\frac{F_x}{10} & 0 & \frac{\ell V_z}{6} & 0 & -\frac{\ell F_x}{30} & \frac{F_x}{10} & 0 & -\frac{\ell V_z}{6} & 0 & \frac{2\ell F_x}{15} \\ \hline
\end{array}\right]
\begin{Bmatrix} u_{ya} \\ u_{za} \\ \theta_{xa} \\ \theta_{ya} \\ \theta_{za} \\ u_{yb} \\ u_{zb} \\ \theta_{xb} \\ \theta_{yb} \\ \theta_{zb} \end{Bmatrix}
\tag{36}
$$

REFERENCE

1. Martin, H. C., "On the Derivation of Stiffness Matrices for the Analysis of Large Deflection and Stability Problems," Proceedings of Conference on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, October 26-28, 1965, Air Force Flight Dynamics Laboratory Report AFFDL TR 66-80, 1966.

522

7.3 PLATES

7.3.1 The Energy of Differential Stiffness

The coordinate system for the basic triangular NASTRAN plate element is shown below.

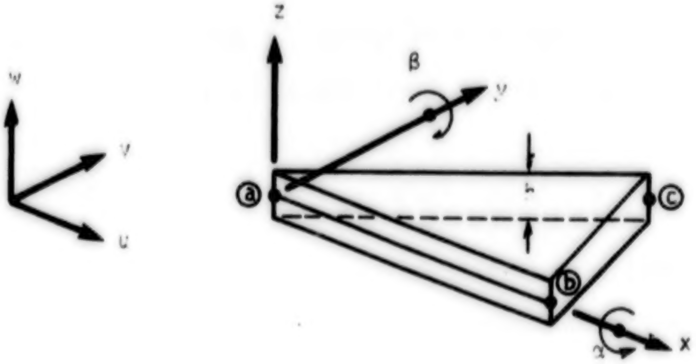


Figure 1. Plate element.

The stress distribution in the plate is:

$$\left.\begin{aligned}
\sigma_z &= 0 \\
\sigma_x &= \bar{\sigma}_x - \frac{M_x z}{I} \\
\sigma_y &= \bar{\sigma}_y - \frac{M_y z}{I} \\
\tau_{xy} &= \bar{\tau}_{xy} - \frac{M_{xy} z}{I} \\
\tau_{xz} &= \frac{V_x}{h} \\
\tau_{yz} &= \frac{V_y}{h}
\end{aligned}\right\} .$$

PAGE 523 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

where

$\bar{\sigma}_x$, $\bar{\sigma}_y$, $\bar{\sigma}_{xy}$ are average in-plane (membrane) stresses ,

M_x, M_y, M_{xy} are bending moments,

V_x, V_y are transverse shear forces,

I is the bending coefficient $\left(\frac{h^3}{12} \text{ for a homogeneous plate}\right)$.

Applying Equation 1 to Equation 16 of Section 7.1 and integrating over the depth of the plate $\left(- \frac{h}{2} < z < \frac{h}{2}\right)$ results in the energy per unit area of the plate.

$$U_A' = \frac{h}{2}\left[\omega_x^2\,\bar{\sigma}_y + \omega_y^2\,\bar{\sigma}_x - 2\omega_x\omega_y\,\bar{\tau}_{xy} + \omega_z^2\,(\bar{\sigma}_x + \bar{\sigma}_y)\right] - \omega_x\omega_z V_x - \omega_y\omega_z V_y \quad . \qquad (2)$$

The last three terms involve rotations about an axis normal to the plane of the plate and they are frequently omitted in buckling analyses. The ω_z^2 term, however, is important for the overall buckling behavior of a built-up beam-like structure and will be retained. The terms proportional to V_x and V_y will be omitted.

The rotations are expressed in terms of displacements as follows:

$$\left.\begin{aligned} \omega_x &= \frac{1}{2}\left[\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right] , \\ \omega_y &= \frac{1}{2}\left[\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right] , \\ \omega_z &= \frac{1}{2}\left[\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right] . \end{aligned}\right\} \qquad (3)$$

Equations 2 and 3 are the basis for calculating differential stiffness of a plate on the assumption that the strains are small compared to rotations. This assumption will not be made. In removing it, the inplane rotations, ω_x and ω_y, will be treated differently than the normal rotation ω_z. The transverse shear strains are

$$\left.\begin{aligned} \gamma_x &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} = \frac{\partial w}{\partial x} + \beta , \\ \gamma_y &= \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} = \frac{\partial w}{\partial x} - \alpha . \end{aligned}\right\} \qquad (4)$$

so that

$$\left.\begin{aligned} \omega_x &= \frac{\partial w}{\partial y} - \frac{1}{2}\gamma_y \ , \\ \omega_y &= -\frac{\partial w}{\partial x} + \frac{1}{2}\gamma_x \ . \end{aligned}\right\} \qquad (5)$$

The transverse shear strains, γ_x and γ_y, are usually small and unimportant. They are, however, important for sandwich plates with stiff face plates and a soft core. For such plates, the rotations ω_x and ω_y should be replaced by $\frac{\partial w}{\partial y}$ and $-\frac{\partial w}{\partial x}$ in Equation 2 as can easily be shown. Consider the term $\frac{h}{2}\omega_x^2\bar{\sigma}_y$ in Equation 2 for a sandwich element that is subjected to simple shear as shown below.

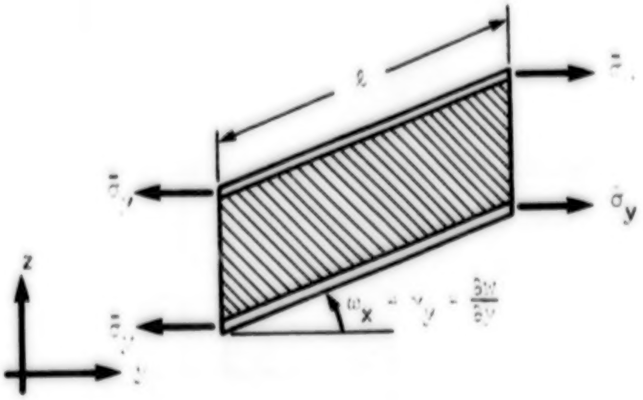


Figure 2. Shear deformation of a sandwich plate.

The core material is subjected to a shear strain, γ_y, and also experiences a volume change, $\Delta V/V = -\frac{1}{2}\gamma_y^2$, which produces tractions tending to increase the length of the face plates. Since the face plates are stiff compared to the core material, they will not change in length and the energy due to differential stiffness is simply

$$\Delta W = \frac{1}{2} A_c \, \ell \, \bar{\sigma}_y \left(\frac{\partial w}{\partial y}\right)^2 \ , \qquad (6)$$

where A_c is the cross-sectional area of the face plate material. It is seen that Equation 5 would give a different result. For sandwich plates ω_x should, therefore, be replaced by $\frac{\partial w}{\partial y}$ in Equation 2 and ω_y should be replaced by $-\frac{\partial w}{\partial x}$. Since the shear strains are small for other types of plates, these substitutions will be used in all cases.

The situation with respect to the normal rotation, ω_z, is much more delicate, and a correct treatment of the effect of strains depends on assumptions regarding the behavior of materials with finite strains. Biot (Reference 1) has developed the following formula for the energy of differential stiffness of compressible materials in plane strain problems.

$$U' = \sigma_x \left[\frac{1}{2}\omega_z^2 + \omega_z \epsilon_{xy} \right] + \sigma_y \left[\frac{1}{2}\omega_z^2 - \omega_z \epsilon_{xy} \right] + \tau_{xy}(\epsilon_y - \epsilon_x)\omega_z \quad , \tag{7}$$

where

$$\left. \begin{aligned} \epsilon_{xy} &= \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) = \frac{1}{2}\gamma \quad , \\ \epsilon_x &= \frac{\partial u}{\partial x} \quad , \\ \epsilon_y &= \frac{\partial v}{\partial y} \quad . \end{aligned} \right\} \tag{8}$$

An independent proof of Equation 7 is as follows. Consider a unit cube under a constant axial tension which undergoes shear strain as shown in Figure 3.

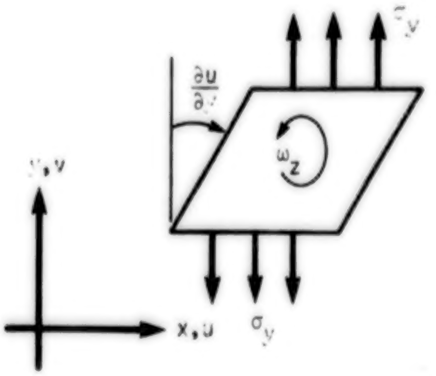


Figure 3. Shear deformation of a unit cube.

The moment about the z axis due to the displaced force is

$$M_z = \sigma_y \frac{\partial u}{\partial y} \quad . \tag{9}$$

The shear strain, $\gamma = 2\epsilon_{xy}$, and the average rotation ω_z, are given by the following equations:

$$\omega_z = \frac{1}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) \quad , \tag{10}$$

$$\gamma = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \quad . \tag{11}$$

Solving Equations 10 and 11:

$$\frac{\partial u}{\partial y} = \frac{1}{2}\gamma - \omega_z \quad , \tag{12}$$

$$\frac{\partial v}{\partial x} = \frac{1}{2}\gamma + \omega_z \quad . \tag{13}$$

The work done due to the moment is

$$W_\sigma = \int_0^{\omega_z} M_z \, d\omega_z \quad . \tag{14}$$

Substituting Equations 9 and 12 into Equation 14 gives

$$W_\sigma = \sigma_y \int_0^{\omega_z} \left(\frac{1}{2}\gamma - \omega_z\right) d\omega_z \quad . \tag{15}$$

Integrating Equation 15 produces

$$W_\sigma = -\frac{1}{2}\sigma_y\left(\omega_z^2 - \gamma\omega_z\right) = -\sigma_y\left(\frac{1}{2}\omega_z^2 - \omega_z\epsilon_{xy}\right) \quad , \tag{16}$$

which is the same as the second term in Equation 7. The terms proportional to σ_x and τ_{xy} in Equation 7 may be similarly derived. In particular the term proportional to τ_{xy} is obtained by considering the equivalent normal stress components in a coordinate system rotated through 45°.

In summary, the expression that is used for the energy of differential stiffness per unit area of a plate element consists of a part, U_b', due to out-of-plane (bending) motions and a part,

U_m', due to in-plane (membrane) motions

$$U_A' = U_b' + U_m' \quad , \tag{17}$$

where

$$U_b' = \frac{h}{2}\left\{\bar{\sigma}_x\left(\frac{\partial w}{\partial x}\right)^2 + \bar{\sigma}_y\left(\frac{\partial w}{\partial y}\right)^2 + 2\bar{\tau}_{xy}\frac{\partial w}{\partial x}\frac{\partial w}{\partial y}\right\} \quad , \tag{18}$$

is the energy for bending motions obtained by substituting $\frac{\partial w}{\partial x}$ and $\frac{\partial w}{\partial y}$ for $-\omega_y$ and ω_x respectively in Equation 2, and where

$$U_m' = \frac{h}{2}\left\{\bar{\sigma}_x(\omega_z^2 + 2\omega_z\epsilon_{xy}) + \bar{\sigma}_y(\omega_z^2 - 2\omega_z\epsilon_{xy}) + 2\bar{\tau}_{xy}(\epsilon_y - \epsilon_x)\omega_z\right\} , \tag{19}$$

is the energy for membrane motions obtained from Equation 7.

The differential stiffness matrices are developed below separately for bending and membrane motions.

7.3.2 Differential Stiffness for the Basic Bending Triangle

The notation and development in Section 5.8.2 will be used without further reference.

The out-of-plane deflections of the basic triangle are described by the polynomial

$$w = w_a + y\alpha_a - x\beta_a + x\gamma_x + y\gamma_y + q_1x^2 + q_2xy + q_3y^2 + q_4x^3 + q_5xy^2 + q_6y^3 , \tag{20}$$

where w_a, α_a and β_a are the out-of-plane displacement and rotation at point (a) in Figure 1.

The linear terms are combined in the equations

$$\left.\begin{aligned} \bar{w}_x &= \gamma_x - \beta_a \quad , \\ \bar{w}_y &= \gamma_y + \alpha_a \quad . \end{aligned}\right\} \tag{21}$$

The slopes of the surface at any point x, y, may be described by the matrix $[H_{\theta q}]$ where:

$$\begin{Bmatrix} \theta_1 \\ \hline \theta_2 \end{Bmatrix} = \begin{Bmatrix} \frac{\partial w}{\partial x} \\ \hline \frac{\partial w}{\partial y} \end{Bmatrix} = [H_{\theta q}(x,y)] \begin{Bmatrix} \bar{w}_x \\ \bar{w}_y \\ \hline q_1 \\ q_2 \\ \vdots \\ q_6 \end{Bmatrix} , \qquad (22)$$

and

$$[H_{\theta q}] = \left[\begin{array}{c|c|c|c|c|c|c|c} 1 & 0 & 2x & y & 0 & 3x^2 & y^2 & 0 \\ \hline 0 & 1 & 0 & x & 2y & 0 & 2xy & 3y^2 \end{array}\right] . \qquad (23)$$

The differential stiffness in terms of the slopes at any point x,y, is given by the equation

$$K_{ij}^{d\theta} = \frac{\partial^2 U_b'}{\partial\theta_i \partial\theta_j} \, dA . \qquad (24)$$

Substituting Equation 18 into Equation 24 results in the matrix:

$$[K^{d\theta}] = h \begin{bmatrix} \bar{\sigma}_x & \bar{\tau}_{xy} \\ \bar{\tau}_{xy} & \bar{\sigma}_y \end{bmatrix} dA . \qquad (25)$$

The differential stiffness for the entire plate in terms of generalized displacements, $\lfloor \bar{w}_x, \bar{w}_y, q_1, q_2, \ldots, q_6 \rfloor$ is the integral:

$$[K^{dq}] = h \int_A [H_{\theta q}]^T [K^{d\theta}] [H_{\theta q}] \, dA . \qquad (26)$$

Explicit formulas for the terms of $[K^{dq}]$ are given in Table 1 in terms of integrals $I_{k\ell}$ defined as

$$I_{k\ell} = h \int_A x^k y^\ell \, dA . \qquad (27)$$

The integrals are easily evaluated by using the natural coordinate functions f_i defined by the equation:

$$r(x,y) = f_1 r_1 + f_2 r_2 + f_3 r_3 \quad , \tag{28}$$

where r is a linear function over the area, and r_1, r_2, r_3 are the values of the function at gridpoints a, b, and c, respectively, in Figure 1.

Since x and y are obviously linear functions we may use the equations:

$$\begin{aligned} x &= f_2 x_2 + f_3 x_3 \quad , \\ y &= f_3 y_3 \quad , \end{aligned} \tag{29}$$

note: $x_1 = y_1 = y_2 = 0$.

A property of these transformations is that:

$$\int_A f_1^\alpha \, f_2^\beta \, f_3^\gamma \, dA = 2A \frac{\alpha!\beta!\gamma!}{(2 + \alpha + \beta + \gamma)!} \quad . \tag{30}$$

Substituting Equation 29 into Equation 27 results in the function:

$$I_{k\ell} = hy_3^\ell \int_A (f_2 x_2 + f_3 x_3)^k \, f_3^\ell \, dA \quad . \tag{31}$$

This expression, with the aid of Equation 30, is used directly to evaluate the integrals. For example,

$$I_{22} = hy_3^2 \int_A [f_2^2 \, f_3^2 \, x_2^2 + 2f_2 \, f_3^3 \, x_2 \, x_3 + f_3^4 \, x_3^2] \, dA \quad . \tag{32}$$

Using Equation 30 for each of the three terms, produces the result

$$I_{22} = 2Ahy_3^2 \left[\frac{2!2!}{6!} x_2^2 + 2 \frac{1!3!}{6!} x_2 x_3 + \frac{1!4!}{6!} x_3^2 \right] = \frac{2Ahy_3}{6!} \left[4x_2^2 + 12x_2 x_3 + 24x_3^2 \right] \tag{33}$$

The remaining task is to define the variables $\bar{w}_x$, $\bar{w}_y$, and $\{q\}$ in terms of the displacements of the connected points. In this connection, the following matrix equations are taken from Section 5.8.2.

$$\{q\} = [H]^{-1}\left\{\{u_i\} - [S]\{u_a\}\right\} , \tag{34}$$

$$\begin{Bmatrix} \gamma_x \\ \gamma_y \end{Bmatrix} = [H_{\gamma q}]\{q\} , \tag{35}$$

where

$$\{u_i\} = \left\{\begin{array}{c} u_b \\ \hline u_c \end{array}\right\} = \left\{\begin{array}{c} w_b \\ \alpha_b \\ \beta_b \\ \hline w_c \\ \alpha_c \\ \beta_c \end{array}\right\} , \tag{36}$$

$$\{u_a\} = \begin{Bmatrix} w_a \\ \alpha_a \\ \beta_a \end{Bmatrix} . \tag{37}$$

In the actual application, three 6 x 3 matrices are generated by multiplication and partitioning as follows:

$$[H_a] = -[H]^{-1}\,[S] , \tag{38}$$

$$[H_b \mid H_c] = [H]^{-1} . \tag{39}$$

The generalized coordinates are then related to gridpoint displacements by

$$\begin{Bmatrix} \bar{w}_x \\ \bar{w}_y \\ q_1 \\ \vdots \\ q_6 \end{Bmatrix} = [C_a]\{u_a\} + [C_b]\{u_b\} + [C_c]\{u_c\} \ , \tag{40}$$

where, using Equations 21, 34, 35, 38, and 39:

$$[C_a] = \left[\begin{array}{c} [H_{\gamma q}]\,[H_a] \\ \\ \hdashline H_a \end{array}\right] + \left[\begin{array}{ccc} 0 & 0 & -1 \\ 0 & 1 & 0 \\ \hdashline & 0 & \end{array}\right] \ , \tag{41}$$

$$[C_b] = \left[\begin{array}{c} [H_{\gamma q}]\,[H_b] \\ \hdashline H_b \end{array}\right] \ , \tag{42}$$

$$[C_c] = \left[\begin{array}{c} [H_{\gamma q}]\,[H_c] \\ \hdashline H_c \end{array}\right] \ . \tag{43}$$

The differential stiffness matrices in terms of the displacements at the corners of the triangle are given by the matrix $[K^{dt}]$, where

$$[K^{dt}] = \left[\begin{array}{c} C_a^T \\ \hdashline C_b^T \\ \hdashline C_c^T \end{array}\right] [K^{dq}]\ [C_a \,\vdots\, C_b \,\vdots\, C_c] \ ,$$

$$= \left[\begin{array}{c:c:c} K^{dt}_{aa} & K^{dt}_{ab} & K^{dt}_{ac} \\ \hdashline K^{dt}_{ba} & K^{dt}_{bb} & K^{dt}_{bc} \\ \hdashline K^{dt}_{ca} & K^{dt}_{cb} & K^{dt}_{cc} \end{array}\right] \ . \tag{44}$$

7.3.3 The Membrane Triangle (TRMEM) and Quadrilateral (QDMEM)

For out-of-plane motion, the membrane triangle is assumed to rotate as a rigid body. The energy is expressed in terms of rotations ω_x, ω_y, ω_z, and strains ε_x, ε_y, and ε_{xy}, assumed constant over the surface of the triangle. From Equations 17, 18, and 19:

$$U = \frac{hA}{2}\left\{ \bar{\sigma}_x \omega_y^2 + \bar{\sigma}_y \omega_x^2 - 2\bar{\tau}_{xy}\omega_x\omega_y + \bar{\sigma}_x (\omega_z^2 + 2\omega_z\varepsilon_{xy}) + \bar{\sigma}_y(\omega_z^2 - 2\omega_z\varepsilon_{xy}) + 2\bar{\tau}_{xy}(\varepsilon_y - \varepsilon_x)\omega_z \right\}, \tag{45}$$

where hA is the volume of the element.

The differential stiffness matrix in terms of the rotations and strains is

$$[K_\omega^d] = \frac{\partial^2 U}{\partial\omega_i \partial\omega_j} , \tag{46}$$

where $\omega_i = \omega_x, \omega_y, \omega_z, \varepsilon_x, \varepsilon_y, \varepsilon_{xy}$.

The rotations and strains are related to corner displacements by

$$\{\omega\} = [C_1^d]\begin{Bmatrix} u_1 \\ v_1 \\ w_1 \end{Bmatrix} + [C_2^d]\begin{Bmatrix} u_2 \\ v_2 \\ w_2 \end{Bmatrix} + [C_3^d]\begin{Bmatrix} u_3 \\ v_3 \\ w_3 \end{Bmatrix} = [C_1^d \,\vdots\, C_2^d \,\vdots\, C_3^d]\begin{Bmatrix} u_1 \\ \hline u_2 \\ \hline u_3 \end{Bmatrix} = [C^d]\{u_e\}. \tag{47}$$

The differential stiffness in terms of the corner displacements is

$$[K_{ee}^d] = [C^d]^T [K_\omega^d][C^d] \tag{48}$$

The elements of matrices $[K_\omega^d]$, $[C_1^d]$, $[C_2^d]$, and $[C_3^d]$ are shown in Table 2.

Quadrilateral elements (QDMEM) are treated as overlapping pairs of triangular elements for the calculation of elastic stiffness, and their differential stiffnesses are similarly treated.

7.3.4 The Triangular Composite Elements (TRIA1 and TRIA2)

For inplane motion, the elements are assumed to have constant rotation and strain. For this purpose, the equations for the triangular membrane, Equations 45 - 48, are used with no contributions from ω_x and ω_y.

For out-of-plane motion, the element is assumed to have the same deflected shape as that assumed in generating the elastic stiffness matrix. The inplane stresses are calculated using the membrane properties. The centroid of the triangle is used to divide the triangle into three subtriangles as shown in Figure 4 below.

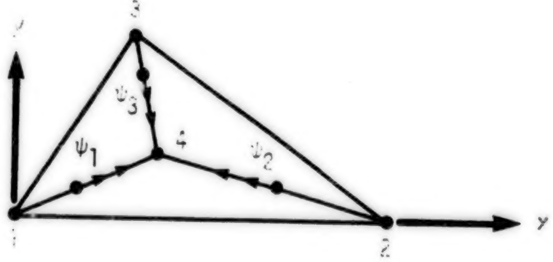


Figure 4. Clough triangle.

The displacement of the center point is constrained to produce compatible slopes, ψ_i, between adjacent triangles, see Section 5.8.3.3.

The calculation procedure is as follows. First, the partitions of Equation 44, are generated for each subtriangle. These matrices are then transformed to the element coordinate system and added to produce $[K_{ij}^{de}]$ where i,j = 1, 2, 3 or 4.

The equations of slope constraint are

$$\begin{Bmatrix} \Delta\psi_1 \\ \Delta\psi_2 \\ \Delta\psi_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} = [G_1]\{u_1\} + [G_2]\{u_2\} + [G_3]\{u_3\} + [G_4]\{u_4\} , \qquad (49)$$

where $\Delta\psi_1$, $\Delta\psi_2$, $\Delta\psi_3$ are the differences in slope between adjacent triangles when point 4 is free, and

$\{u_1\} = \begin{Bmatrix} u_{z1} \\ \theta_{x1} \\ \theta_{y1} \end{Bmatrix}$, etc., are the displacements of the corner points in the element coordinate system. See Section 5.8.3.3 for details.

Applying the equations of constraint to eliminate the elements of $\{u_4\}$ results in the differential stiffness matrix of the element, $[K^d_{ij}]$, referred to the vertices of the composite triangle. Each partition of the matrix corresponding to points i and j is:

$$[K^d_{ij}] = [K^{de}_{ij}] - [K^{de}_{4i}]^T[G_4]^{-1}[G_j] - ([G_4{}^{-1}][G_i])^T[K^{de}_{4j}] + ([G_4]^{-1}[G_i])^T[K^{de}_{44}][G_4]^{-1}[G_j] \tag{50}$$

7.3.5 The Quadrilateral Composite Elements (QUAD1 and QUAD2)

These elements are composed of four overlapping triangles. The inplane differential stiffness of each triangle is calculated with the same equations as the membrane triangle, Section 7.3.3, except that the ω_x and ω_y terms are ignored. For out-of-plane motions, the element is assumed to lie in a plane, parallel to, and midway between the diagonals. The corners of the element are connected to the four grid points by rigid bars. The inplane stresses σ_x, σ_y, and τ_{xy} are calculated independently for each triangle and the differential stiffness is calculated using the equations given in Section 7.3.2. The differential stiffness matrix terms for each triangle are then transformed to the element coordinate system and added.

REFERENCE

1. Biot, M. A., Mechanics of Incremental Deformation, Wiley and Sons, 1965, p. 136.

Table 1 Elements of $[K^{dq}]$

The elements of the (8 x 8) differential stiffness matrix $[K^{dq}]$ are evaluated below from Equation 26. The matrix is symmetric so only the upper triangle terms are given. The superscript $(^{dq})$ is omitted for convenience.

$K_{11} = \sigma_x I_{00}$

$K_{12} = \tau I_{00}$

$K_{13} = 2\sigma_x I_{10}$

$K_{14} = \sigma_x I_{01} + \tau I_{10}$

$K_{15} = 2\tau I_{01}$

$K_{16} = 3\sigma_x I_{20}$

$K_{17} = \sigma_x I_{02} + 2\tau I_{11}$

$K_{18} = 3\tau I_{02}$

$K_{22} = \sigma_y I_{00}$

$K_{23} = 2\tau I_{10}$

$K_{24} = \tau I_{01} + \sigma_y I_{10}$

$K_{25} = 2\sigma_y I_{01}$

$K_{26} = 3\tau I_{20}$

$K_{27} = \tau I_{02} + 2\sigma_y I_{11}$

$K_{28} = 3\sigma_y I_{02}$

$K_{33} = 4\sigma_x I_{20}$

$K_{34} = 2(\sigma_x I_{11} + \tau I_{20})$

$K_{35} = 4\tau I_{11}$

$K_{36} = 6\sigma_x I_{30}$

$K_{37} = 2(\sigma_x I_{12} + 2\tau I_{21})$

$K_{38} = 6\tau I_{12}$

$K_{44} = \sigma_x I_{02} + 2\tau I_{11} + \sigma_y I_{20}$

$K_{45} = 2(\tau I_{02} + \sigma_y I_{11})$

$K_{46} = 3(\sigma_x I_{21} + \tau I_{30})$

$K_{47} = \sigma_x I_{03} + 3\tau I_{12} + 2\sigma_y I_{21}$

$K_{48} = 3(\tau I_{03} + \sigma_y I_{12})$

$K_{55} = 4\sigma_y I_{02}$

$K_{56} = 6\tau I_{21}$

$K_{57} = 2(\tau I_{03} + 2\sigma_y I_{12})$

$K_{58} = 6\sigma_y I_{03}$

$K_{66} = 9\sigma_x I_{40}$

$K_{67} = 3(\sigma_x I_{22} + 2\tau I_{31})$

$K_{68} = 9\tau I_{22}$

$K_{77} = \sigma_x I_{04} + 4\tau I_{13} + 4\sigma_y I_{22}$

$K_{78} = 3(\tau I_{04} + 2\sigma_y I_{13})$

$K_{88} = 9\sigma_y I_{04}$

Table 2 Matrices for the TRMEM Element.

$$[K_\omega^d] = At\begin{bmatrix} \sigma_y & -\sigma_{xy} & 0 & 0 & 0 & 0 \\ -\sigma_{xy} & \sigma_x & 0 & 0 & 0 & 0 \\ 0 & 0 & (\sigma_x+\sigma_y) & -\tau_{xy} & \tau_{xy} & (\sigma_x-\sigma_y) \\ 0 & 0 & -\tau_{xy} & 0 & 0 & 0 \\ 0 & 0 & \tau_{xy} & 0 & 0 & 0 \\ 0 & 0 & (\sigma_x-\sigma_y) & 0 & 0 & 0 \end{bmatrix}$$

The corresponding degrees of freedom are: ω_x, ω_y, ω_z, ε_x, ε_y, and ε_{xy}.)

$$[C_1^d] = \begin{bmatrix} 0 & 0 & \gamma_3-\gamma_2 \\ 0 & 0 & \gamma_1 \\ \frac{\gamma_2-\gamma_3}{2} & -\frac{\gamma_1}{2} & 0 \\ -\gamma_1 & 0 & 0 \\ 0 & \gamma_3-\gamma_2 & 0 \\ -\frac{\gamma_2-\gamma_3}{2} & -\frac{\gamma_1}{2} & 0 \end{bmatrix}$$

$$[C_2^d] = \begin{bmatrix} 0 & 0 & -\gamma_3 \\ 0 & 0 & -\gamma_1 \\ \frac{\gamma_3}{2} & \frac{\gamma_1}{2} & 0 \\ \gamma_1 & 0 & 0 \\ 0 & -\gamma_3 & 0 \\ -\frac{\gamma_3}{2} & \frac{\gamma_1}{2} & 0 \end{bmatrix}$$

$$[C_3^d] = \begin{bmatrix} 0 & 0 & \gamma_2 \\ 0 & 0 & 0 \\ -\frac{\gamma_2}{2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ \frac{\gamma_2}{2} & 0 & 0 \end{bmatrix}$$

where

$$\gamma_1 = \frac{1}{x_2}$$

$$\gamma_2 = \frac{1}{y_3}$$

$$\gamma_3 = \frac{x_3}{x_2 y_3}$$

7.4 THE CONICAL SHELL ELEMENT

The elastic stiffness matrix for the conical shell element is derived in Section 5.9. The differential stiffness matrix due to static axisymmetric preload is derived here using the notation and the results of Section 5.9 and the general theory of differential stiffness developed in Section 7.1.

7.4.1 General Method

The static preload used to generate differential stiffness must be axisymmetric because an unsymmetrical distribution of preload provides an unsymmetrical distribution of differential stiffness, thus violating the symmetry assumption of the conical shell element. Furthermore, the effects of static bending moments and transverse shears on differential stiffness will be neglected, which is the usual practice in the analysis of the buckling of shells, leaving only the direct membrane stresses, σ_{so} and $\sigma_{\phi o}$, of zero harmonic order as contributors to differential stiffness. The stiffening effect of membrane shear stress of zero harmonic order, $\sigma_{s\phi o}$, is not axisymmetric, since shells that have been subjected to a clockwise twist can be distinguished from those that have been subjected to a counter-clockwise twist.

The potential energy of differential stiffness for a conical shell element can be written as follows, using the result given in Equation 14 of Section 7.1.

$$V_d = \frac{1}{2}\int_0^{\ell}\int_0^{2\pi}\left[F_{so}\omega_\phi^{\,2} + F_{\phi o}\omega_s^{\,2} + (F_{so} + F_{\phi o})\omega_w^{\,2}\right] r\,d\phi\,ds \quad . \qquad (1)$$

F_{so} and $F_{\phi o}$ are the direct membrane forces per unit width of zero harmonic order in the meridional and azimuthal directions. ω_ϕ, ω_s and ω_w are rotations about the orthogonal axes defined in Figure 1 of Section 5.9. The rotations ω_ϕ and ω_s should be computed from translational motions normal to the neutral surface rather than from the tangential motions of points on the inner and outer surfaces. They are, therefore, equal to the quantities α and β in Equations 37 and 38 of Section 5.9, with the transverse shear strains, γ_s and γ_d, set equal to zero. ω_w is identical to θ_w given by Equation 42 of Section 5.9. Thus the rotations are related to translational displacements by

PAGE 541 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

$$\omega_\phi = \frac{\partial w}{\partial s} \quad ,$$

$$\omega_s = -\frac{1}{r}\frac{\partial w}{\partial \phi} + \frac{\cos\psi}{r} u \quad , \qquad (2)$$

$$\omega_w = -\frac{1}{2}\left[\frac{\partial u}{\partial s} + \frac{u\sin\psi}{r} - \frac{1}{r}\frac{\partial v}{\partial \phi}\right] \quad .$$

The rotations have the following harmonic dependence

$$\omega_\phi = \sum_{n=1}^{m} \omega_{\phi n} \cos n\phi + \omega_{\phi o} + \sum_{n=1}^{m} \omega^*_{\phi n} \sin n\phi \quad ,$$

$$\omega_s = \sum_{n=1}^{m} \omega_{sn} \sin n\phi + \omega^*_{so} - \sum_{n=1}^{m} \omega^*_{sn} \cos n\phi \quad , \qquad (3)$$

$$\omega_w = \sum_{n=1}^{m} \omega_{wn} \sin n\phi + \omega^*_{wo} - \sum_{n=1}^{m} \omega^*_{wn} \cos n\phi \quad .$$

As with all other displacements and strains, the harmonic coefficients of different harmonic orders are uncoupled, and the starred and unstarred coefficients of the same harmonic order are uncoupled. By virtue of these facts, Equations 2 can be written for each harmonic coefficient as follows

$$\omega_{\phi n} = \frac{\partial w_n}{\partial s} \quad ,$$

$$\omega_{sn} = \frac{n}{r} w_n + \frac{\cos\psi}{r} u_n \quad , \qquad (4)$$

$$\omega_{wn} = -\frac{1}{2}\left[\frac{\partial u_n}{\partial s} + \frac{\sin\psi}{r} u_n + \frac{n}{r} v_n\right] \quad ,$$

and similarly for the starred coefficients.

The potential energy of differential stiffness can be separated into independent terms

$$V_d = V_{do} + \sum_{n=1}^{m} V_{dn} + V^*_{do} + \sum_{n=1}^{m} V^*_{dn} \quad , \qquad (5)$$

ORIGINAL PAGE IS OF POOR QUALITY

where, performing the integration with respect to ϕ

$$V_{do} = \pi \int_0^{\ell} F_{so}\omega^2_{\phi o} r ds \quad , \tag{6}$$

$$V^{\star}_{do} = \pi \int_0^{\ell} \left[F_{\phi o}\omega^{\star 2}_{so} + (F_{so} + F_{\phi o})\omega^{\star 2}_{wo} \right] rds \quad , \tag{7}$$

$$V_{dn} = \frac{\pi}{2} \int_0^{\ell} \left[F_{so}\omega_{\phi n}^2 + F_{\phi o}\omega_{sn}^2 + (F_{so} + F_{\phi o})\omega_{wn}^2 \right] rds \quad . \tag{8}$$

The form for $V^{\star}_{dn}$ is identical to that for V_{dn} with the substitution of starred coefficients.

It is convenient to define the following diagonal matrix

$$[F_o] = \begin{bmatrix} F_{so} & 0 & 0 \\ 0 & F_{\phi o} & 0 \\ 0 & 0 & F_{so} + F_{\phi o} \end{bmatrix} , \tag{9}$$

and also

$$\{\omega_n\} = \begin{Bmatrix} \omega_{\phi n} \\ \omega_{sn} \\ \omega_{wn} \end{Bmatrix} , \tag{10}$$

$$\{\omega_o\} = \begin{Bmatrix} \omega_{\phi o} \\ \omega^{\star}_{so} \\ \omega^{\star}_{wo} \end{Bmatrix} , \tag{11}$$

so that Equations 6, 7, and 8 can be written as

$$V_{do} + V^{\star}_{do} = \pi \int_0^{\ell} \{\omega_o\}^T [F_o]\{\omega_o\} \; rds \quad , \tag{12}$$

and

$$V_{dn} = \frac{\pi}{2}\int_0^{\ell}\{\omega_n\}^T[F_o]\{\omega_n\}rds \qquad n > 0 \quad . \tag{13}$$

The Fourier components of rotation are related to the independent constants $\{q_n\}$ by

$$\{\omega_n\} = [H_{\omega q}]_n\{q_n\} \qquad n \geq 0 \quad , \tag{14}$$

so that

$$V_{do} + V_{do}^{*} = \frac{1}{2}\{q_o\}^T[K_o^{qd}]\{q_o\} \quad , \tag{15}$$

$$V_{dn} = \frac{1}{2}\{q_n\}^T[K_n^{qd}]\{q_n\} \quad , \tag{16}$$

and

$$V_{dn}^{*} = \frac{1}{2}\{q_n^{*}\}^T[K_n^{qd}]\{q_n^{*}\} \quad , \tag{17}$$

where

$$[K_o^{qd}] = 2\pi\int_0^{\ell}[H_{\omega q}]_o^T[F_o][H_{\omega q}]_n\ rds \quad , \tag{18}$$

and

$$[K_n^{qd}] = \pi\int_0^{\ell}[H_{\omega q}]_n^T[F_o][H_{\omega q}]_n\ rds \quad , \tag{19}$$

are the differential stiffness matrices referred to the independent constants. (Note that the differential stiffness matrices for starred and unstarred coefficients are identical.) The differential stiffness matrices referred to the degrees of freedom for the element, $\{u_e\}$, are

$$[K_n^{ud}] = [H_{uq}^{-1}]_n^T[K_n^{qd}][H_{uq}^{-1}]_n \qquad n \geq 0 \quad . \tag{20}$$

Formulas for evaluation of the elements of $[H_{uq}]$ are given in Section 5.9.

The remaining task is to derive formulas for the elements of $[K_n^{qd}]$. The formulas for $[K_o^{qd}]$ differ only by a factor of two, (compare Equations 18 and 19).

7.4.2 Evaluation of $[K_n^{qd}]$

The relationship between the harmonic coefficients of displacement and the independent constants is stated in Equations 30, 31, and 32 of Section 5.9. Substituting into Equation 4 and suppressing the subscript (n), we find

$$
\begin{aligned}
\omega_\phi &= q_6 + 2q_7 s + 3q_8 s^2 \quad , \\
\omega_s &= \frac{n}{r}\left(q_5 + q_6 s + q_7 s^2 + q_8 s^3\right) + \frac{\cos\psi}{r}\left(q_1 + q_2 s\right) \quad , \qquad (21) \\
\omega_w &= -\frac{1}{2}\left(q_2 + \frac{\sin\psi}{r}\left(q_1 + q_2 s\right) + \frac{n}{r}\left(q_3 + q_4 s\right)\right) \quad .
\end{aligned}
$$

The coefficients in these equations are the elements of the $[H_{\omega q}]$ matrix, written below in transposed form

$$
[H_{\omega q}]^T = \begin{array}{cc}
\left[\begin{array}{ccc}
0 & \cos\psi/r & -\sin\psi/2r \\
0 & s\cos\psi/r & -\frac{1}{2} - s\cdot\sin\psi/2r \\
0 & 0 & -n/2r \\
0 & 0 & -ns/2r \\
0 & n/r & 0 \\
1 & ns/r & 0 \\
2s & ns^2/r & 0 \\
3s^2 & ns^3/r & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{array}\right] &
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10
\end{array} \\
\begin{array}{ccc}
\omega_\phi & \omega_s & \omega_w
\end{array} &
\end{array} \quad . \qquad (22)
$$

Turning now to the evaluation of the elements of $[F_o]$, Equation 9, it is required to find the explicit dependence of the elements of $[F_o]$ on distance, s, along the conical segment. From Equation 17 of Section 5.9:

$$F_{so} = t[E_{11}(\varepsilon_{so} - \Delta\varepsilon_{so}) + E_{12}(\varepsilon_{\phi o} - \Delta\varepsilon_{\phi o})] \quad ,$$

$$F_{\phi o} = t[E_{12}(\varepsilon_{so} - \Delta\varepsilon_{so}) + E_{22}(\varepsilon_{\phi o} - \Delta\varepsilon_{\phi o})] \quad , \qquad (23)$$

where the $\Delta\varepsilon$'s are enforced strains including those produced by temperature.

From Equations 60 and 61 of Section 5.9, with $n = 0$,

$$\varepsilon_{so} = q_{40} \quad ,$$

$$\varepsilon_{\phi o} = \frac{1}{2}\left[\sin\psi(q_{30} + q_{40}s) + \cos\psi(q_{50} + q_{60}s + q_{70}s^2 + q_{80}s^3)\right] \quad . \qquad (24)$$

Formulas for the $\Delta\varepsilon$'s are given in Equation 123 of Section 5.9. Thus, F_{so} and $F_{\phi o}$ may be written

$$F_{so} = a_o + a_1s + a_2s^2 + a_3s^3 \quad ,$$

$$F_{\phi o} = b_o + b_1s + b_2s^2 + b_3s^3 \quad , \qquad (25)$$

where

$$a_o = t[E_{11}q_{40} + \frac{1}{r}E_{12}(\sin\psi q_{30} + \cos\psi q_{50})] + (\text{part due to } \Delta\varepsilon\text{'s}) \quad ,$$

$$a_1 = \frac{1}{r}tE_{12}(\sin\psi q_{40} + \cos\psi q_{60}) + (\text{part due to } \Delta\varepsilon\text{'s}) \quad , \qquad (26)$$

$$a_2 = \frac{1}{r}tE_{12}\cos\psi q_{70} \quad ,$$

$$a_3 = \frac{1}{r}tE_{12}\cos\psi q_{80} \quad ,$$

$$b_o = t[E_{12}q_{40} + \frac{1}{r}E_{22}(\sin\psi q_{30} + \cos\psi q_{50})] + (\text{part due to } \Delta\varepsilon\text{'s}) \quad ,$$

$$b_1 = \frac{1}{r}tE_{22}(\sin\psi q_{40} + \cos\psi q_{60}) + (\text{part due to } \Delta\varepsilon\text{'s}) \quad ,$$

$$b_2 = \frac{1}{r}tE_{22}\cos\psi q_{70} \quad , \qquad (27)$$

$$b_3 = \frac{1}{r}tE_{22}\cos\psi q_{80} \quad ,$$

and q_{30}--q_{80} are, of course, obtained as part of the stress data reduction procedure for the preceding static solution, see Section 5.9.10.

The elements of $[K_n^{qd}]$ will now be evaluated from Equation 19, carrying out the indicated integration. For the purpose it is convenient to define the integrals

$$A_{mn} = \pi \int_0^{\ell} s^m r^{1-n} F_{so} ds \quad ,$$

$$B_{mn} = \pi \int_0^{\ell} s^m r^{1-n} F_{\phi o} ds \quad , \qquad (28)$$

$$C_{mn} = A_{mn} + B_{mn} \quad .$$

A_{mn} and B_{mn} can be evaluated in terms of the integrals I_{mn} defined in Equation 82 of Section 5.9, and the coefficients defined in Equations 26 and 27.

$$A_{mn} = a_o I_{mn} + a_1 I_{m+1,n} + a_2 I_{m+2,n} + a_3 I_{m+3,n} \quad ,$$

$$B_{mn} = b_o I_{mn} + b_1 I_{m+1,n} + b_2 I_{m+2,n} + b_3 I_{m+3,n} \quad . \qquad (29)$$

The evaluation of the (i,j) element of $[K_n^{qd}]$ consists of multiplying the ith row of Equation 22 by the jth row, with the weighting factors A_{mn}, B_{mn}, and C_{mn} applied to the terms from the first, second, and third columns respectively. m is equal to the power of s appearing in the numerator of the product, and n is equal to the power of r occurring in the denominator of the product. The resulting matrix is symmetric. Explicit formulas for the elements will be found in Section 4.87 of the Programmer's Manual.

548

ORIGINAL PAGE IS OF POOR QUALITY

7.5 ISOPARAMETRIC HEXAHEDRON THREE-DIMENSIONAL SOLID ELEMENTS

The differential stiffness matrix for the isoparametric solid elements is obtained by adding the energy due to initial stresses to the potential energy function. This additional energy is derived in Section 7.1, and is given by

$$w_p = \frac{1}{2}\int_V [\omega_x^2(\sigma_y + \sigma_z) + \omega_y^2(\sigma_x + \sigma_z) + \omega_z^2(\sigma_x + \sigma_y) - 2\omega_x\,\omega_y\,\tau_{xy} - 2\omega_y\,\omega_z\,\tau_{yz} - 2\omega_z\,\omega_x\,\tau_{zx}]dV\,, \quad (1)$$

where the rotations are given by the relations

$$\left.\begin{aligned} \omega_x &= \frac{1}{2}\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right) \\ \omega_y &= \frac{1}{2}\left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right) \\ \omega_z &= \frac{1}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) \end{aligned}\right\}\,, \quad (2)$$

and may be expressed in terms of the grid point displacements by using Equation 1 of Section 5.13.1, i.e.,

$$\begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = [\bar{C}_1 \mid \bar{C}_2 \mid \bar{C}_3 \mid \ldots \bar{C}_{NGP}] \begin{Bmatrix} u_1 \\ v_1 \\ w_1 \\ \hdashline u_2 \\ v_2 \\ w_2 \\ \hdashline \vdots \\ \hdashline u_{NGP} \\ v_{NGP} \\ w_{NGP} \end{Bmatrix} = [\bar{C}]\{u_e\} \quad , \quad (3)$$

and

$$\bar{C}_i = \begin{bmatrix} 0 & -\dfrac{\partial N_i}{\partial z} & \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial z} & 0 & -\dfrac{\partial N_i}{\partial x} \\ -\dfrac{\partial N_i}{\partial y} & \dfrac{\partial N_i}{\partial x} & 0 \end{bmatrix} \quad .$$

PAGE 549 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

The substitution of Equation 3 into Equation 1 and the addition of this function to the potential energy expression yields the differential stiffness matrix:

$$[K^d_{ee}] = \int_V [\bar{C}]^T [K^d_{\omega\omega}] [\bar{C}] \, dV \quad , \tag{4}$$

where

$$[K^d_{\omega\omega}] = \begin{bmatrix} \sigma_y + \sigma_z & -\tau_{xy} & -\tau_{zx} \\ -\tau_{xy} & \sigma_x + \sigma_z & -\tau_{zy} \\ -\tau_{zx} & -\tau_{zy} & \sigma_x + \sigma_y \end{bmatrix} \quad . \tag{5}$$

As in the structural stiffness matrix, the evaluation of the integral in Equation 4 is obtained by application of the Gaussian Quadrature Formula (see Table 2).

7.6 THE TRIANGULAR THIN SHALLOW SHELL ELEMENT

The expression that is used for the energy of differential stiffness per unit area of the triangular thin shallow shell element consists of a part U_b' due to out-of-plane motions and a part U_m' due to in-plane motions. The expressions for U_b' and U_m' are the same as those for plate elements and are given by Equations 18 and 19 of Section 7.3.1; the expressions for the membrane strains, however, involve the effects of coupling due to bending. Thus,

$$U = U_b' + U_m' \quad , \tag{1}$$

where

$$U_b' = \frac{t}{2} \left\{ \bar{\sigma}_x \left(\frac{\partial w}{\partial x}\right)^2 + \bar{\sigma}_y \left(\frac{\partial w}{\partial y}\right)^2 + 2\bar{\tau}_{xy} \frac{\partial w}{\partial x} \frac{\partial w}{\partial y} \right\} \tag{2}$$

and

$$U_m' = \frac{t}{2} \left\{ \bar{\sigma}_x (\omega_z^2 + 2\omega_z \, \varepsilon_{xy}) + \bar{\sigma}_y (\omega_z^2 - 2\omega_z \, \varepsilon_{xy}) + 2\bar{\tau}_{xy} (\varepsilon_y - \varepsilon_x) \, \omega_z \right\} \quad . \tag{3}$$

The stresses $\bar{\sigma}_x$, $\bar{\sigma}_y$, and $\bar{\tau}_{xy}$ at any point within the element are assumed to vary linearly, the values at the three corner grid points being used to evaluate the coefficients in the linear variation. Thus,

$$\bar{\sigma}_x \, (x,y) = e_1 + e_2 x + e_3 y \quad , \tag{4}$$

$$\bar{\sigma}_y \, (x,y) = f_1 + f_2 x + f_3 y \quad , \tag{5}$$

and

$$\bar{\sigma}_{xy} \, (x,y) = g_1 + g_2 x + g_3 y \quad . \tag{6}$$

In condensed form,

$$\bar{\sigma}_x = \sum_{i=1}^{3} e_i \, x^{R_i} \, y^{S_i} \quad , \tag{7}$$

$$\bar{\sigma}_y = \sum_{i=1}^{3} f_i \, x^{R_i} \, y^{S_i} \quad , \tag{8}$$

and

$$\bar{\sigma}_{xy} = \sum_{i=1}^{3} g_i \, x^{R_i} \, y^{S_i} \quad . \tag{9}$$

Also,

$$\omega_x = \frac{\partial w}{\partial y} \quad , \tag{10}$$

$$\omega_y = -\frac{\partial w}{\partial x} \quad , \tag{11}$$

$$\omega_z = \frac{1}{2}\left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) \quad , \tag{12}$$

$$\varepsilon_x = \frac{\partial u}{\partial x} - z,_{xx}\, w \quad , \tag{13}$$

$$\varepsilon_y = \frac{\partial v}{\partial y} - z,_{yy}\, w \quad , \tag{14}$$

and

$$\varepsilon_{xy} = \frac{\partial v}{\partial x} + \frac{\partial w}{\partial y} - 2z,_{xy}\, w \quad . \tag{15}$$

The thickness t of the element at any point (x,y) is given by

$$t(x,y) = \sum_{i=1}^{3} d_k\, x^{t_k}\, x^{u_k} \quad . \tag{16}$$

The j^{th} column of the i^{th} row of the generalized differential stiffness matrix is

$$\begin{aligned}
k_{ij} = \sum_{k=1}^{3}\sum_{\ell=1}^{3} [&d_k\, e_\ell\, r_i\, r_j\, F(r_i + r_j + t_k + R_\ell - 2,\ s_i + s_j + u_k + S_\ell) \\
&+ d_k\, f_\ell\, s_i\, s_j\, F(r_i + r_j + t_k + R_\ell,\ s_i + s_j + u_k + S_\ell - 2) \\
&+ d_k\, g_\ell\, s_i\, r_j\, F(r_i + r_j + t_k + R_\ell - 1,\ s_i + s_j + u_k + S_\ell - 1) \\
&+ d_k\, g_\ell\, s_j\, r_i\, F(r_i + r_j + t_k + R_\ell - 1,\ s_i + s_j + u_k + S_\ell - 1) \\
&+ 0.25\, d_k\, e_\ell\, p_i\, p_j\, F(p_i + p_j + t_k + R_\ell - 2,\ q_i + q_j + u_k + S_\ell) \\
&+ 0.25\, d_k\, e_\ell\, n_i\, n_j\, F(m_i + m_j + t_k + R_\ell,\ n_i + n_j + u_k + S_\ell - 2) \\
&- 0.25\, d_k\, e_\ell\, p_i\, n_j\, F(p_i + m_j + t_k + R_\ell - 1,\ q_i + n_j + u_k + S_\ell - 1) \\
&- 0.25\, d_k\, e_\ell\, n_i\, p_j\, F(m_i + p_j + t_k + R_\ell - 1,\ n_i + q_j + u_k + S_\ell - 1) \\
&+ d_k\, e_\ell\, p_i\, n_j\, F(p_i + m_j + t_k + R_\ell - 1,\ q_i + n_j + u_k + S_\ell - 1) \\
&+ d_k\, e_\ell\, p_i\, p_j\, F(p_i + p_j + t_k + R_\ell - 2,\ q_i + q_j + u_k + S_\ell) \\
&- d_k\, e_\ell\, p_i\, h_5\, F(p_i + r_j + t_k + R_\ell - 1,\ q_i + s_j + u_k + S_\ell)
\end{aligned} \tag{17}$$

(continued)

$$- d_k e_\ell n_i n_j F(m_i + m_j + t_k + R_\ell, n_i + n_j + u_k + S_\ell - 2)$$

$$- d_k e_\ell n_i p_j F(m_i + p_j + t_k + R_\ell - 1, n_i + q_j + u_k + S_\ell - 1)$$

$$+ d_k e_\ell n_i h_5 F(m_i + r_j + t_k + R_\ell, n_i + s_j + u_k + S_\ell - 1)$$

$$+ 0.25 d_k f_\ell p_i p_j F(p_i + p_j + t_k + R_\ell - 2, q_i + q_j + u_k + S_\ell)$$

$$- 0.25 d_k f_\ell p_i n_j F(p_i + m_j + t_k + R_\ell - 1, q_i + n_j + u_k + S_\ell - 1)$$

$$+ 0.25 d_k f_\ell n_i n_j F(m_i + m_j + t_k + R_\ell, n_i + n_j + u_k + S_\ell - 2)$$

$$- 0.25 d_k f_\ell n_i p_j F(m_i + p_j + t_k + R_\ell - 1, n_i + q_j + u_k + S_\ell - 1)$$

$$- d_k f_\ell p_i n_j F(p_i + m_j + t_k + R_\ell - 1, q_i + n_j + u_k + S_\ell - 1)$$

$$- d_k f_\ell p_i p_j F(p_i + p_j + t_k + R_\ell - 2, q_i + q_j + u_k + S_\ell)$$

$$+ d_k f_\ell p_i h_5 F(p_i + r_j + t_k + R_\ell - 1, q_i + s_j + u_k + S_\ell)$$

$$+ d_k f_\ell n_i n_j F(m_i + m_j + t_k + R_\ell, n_i + n_j + u_k + S_\ell - 2)$$

$$+ d_k f_\ell n_i p_j F(m_i + p_j + t_k + R_\ell - 1, n_i + q_j + u_k + S_\ell - 1)$$

$$- d_k f_\ell n_i h_5 F(m_i + r_j + t_k + R_\ell, n_i + s_j + u_k + S_\ell - 1)$$

$$+ 0.5 d_k g_\ell q_i p_j F(p_i + p_j + t_k + R_\ell - 1, q_i + q_j + u_k + S_\ell - 1)$$

$$+ 0.5 d_k g_\ell q_j p_i F(p_i + p_j + t_k + R_\ell - 1, q_i + q_j + u_k + S_\ell - 1)$$

$$- 0.5 d_k g_\ell q_i n_j F(p_i + m_j + t_k + R_\ell, q_i + n_j + u_k + S_\ell - 2)$$

$$- 0.5 d_k g_\ell q_j n_i F(p_j + m_i + t_k + R_\ell, q_j + n_i + u_k + S_\ell - 2)$$

$$- 0.5 d_k g_\ell p_j h_6 F(v_i + p_j + t_k + R_\ell - 1, s_i + q_j + u_k + S_\ell)$$

$$- 0.5 d_k g_\ell p_i h_6 F(v_j + p_i + t_k + R_\ell - 1, s_j + q_i + u_k + S_\ell)$$

$$+ 0.5 d_k g_\ell n_j h_6 F(r_i + m_j + t_k + R_\ell, s_i + n_j + u_k + S_\ell - 1)$$

$$+ 0.5 d_k g_\ell n_i h_6 F(r_j + m_i + t_k + R_\ell, s_j + n_i + u_k + S_\ell - 1)$$

$$- 0.5 d_k g_\ell m_i p_j F(m_i + p_j + t_k + R_\ell - 2, n_i + q_j + u_k + S_\ell)$$

$$- 0.5 d_k g_\ell m_j p_i F(m_j + p_i + t_k + R_\ell - 2, n_j + q_i + u_k + S_\ell) \qquad (17)$$

(continued)

$$
\begin{aligned}
&+ 0.5\, d_k\, g_\ell\, m_i\, n_j\, F(m_i + m_j + t_k + R_\ell - 1,\ n_i + n_j + u_k + S_\ell - 1) \\
&+ 0.5\, d_k\, g_\ell\, m_j\, n_i\, F(m_i + m_j + t_k + R_\ell - 1,\ n_i + n_j + u_k + S_\ell - 1) \\
&+ 0.5\, d_k\, g_\ell\, p_j\, h_4\, F(r_i + p_j + t_k + R_\ell - 1,\ s_i + q_j + u_k + S_\ell) \\
&+ 0.5\, d_k\, g_\ell\, p_i\, h_4\, F(r_j + p_i + t_k + R_\ell - 1,\ s_j + q_i + u_k + S_\ell) \\
&- 0.5\, d_k\, g_\ell\, n_j\, h_4\, F(r_i + m_j + t_k + R_\ell,\ s_i + n_j + u_k + S_\ell - 1) \\
&- 0.5\, d_k\, g_\ell\, n_i\, h_4\, F(r_j + m_i + t_k + R_\ell,\ s_j + n_i + u_k + S_\ell - 1)]
\end{aligned}
\qquad (17)
$$

(concluded)

The generalized differential stiffness matrix is transformed to the element coordinate system, the basic coordinate system and the global coordinate system in the normal manner.

8. HEAT TRANSFER ANALYSIS

8.1 GENERAL FEATURES

An analogy between thermodynamics and the mechanics of solid bodies has been exploited to extend the capability of NASTRAN to heat transfer analysis. As in the case of structural analysis, the analysis of heat transfer in a solid continuum can be reduced by finite element techniques (see, for instance, Reference 1) to the solution of a set of equilibrium equations in which the unknowns are defined at a discrete set of points. Thus, the general equation that is solved when finite element methods are applied to heat transfer analysis may be written in the form

$$[K]\{u\} + [B]\{\dot{u}\} = \{P\} + \{N\}, \qquad (1)$$

where

$\{u\}$ is a vector of temperatures at gridpoints

$\{P\}$ is a vector of applied heat flows that are known as functions of time

$\{N\}$ is a vector of nonlinear heat flows that depend on temperature

$[K]$ is a symmetric matrix of constant heat conduction coefficients

$[B]$ is a symmetric matrix of constant heat capacity coefficients.

The symbols used in Equation 1 have been deliberately chosen to coincide with some of the structural analysis symbols defined in Table 1 on Page 3.3-4, thereby defining the thermo-mechanical analogy. Heat transfer analysis with NASTRAN uses all of the normal analytical tools provided for structural analysis, the difference being that the arrays $[K]$, $[B]$, $\{P\}$ and $\{N\}$ are computed from thermodynamic properties, rather than from structural properties.

Gridpoints are used to locate temperatures just as they are used to locate displacements in structural analysis. However, one of the major differences between thermodynamics and mechanics is that temperature is a scalar function of position, whereas displacement is a vector which NASTRAN assumes may have as many as six components. Thus, in heat transfer analysis, NASTRAN provides only one degree of freedom at each gridpoint.

556

The heat conduction matrix, [K] and the heat capacity matrix, [B], are formed from "element" properties, just as in structural analysis. Volume heat conduction "elements" are analogous in many ways to structural elements and they even use the same connection and property cards. In addition, a part of the heat conduction matrix may be associated with surface heat convection or radiation. The theory for NASTRAN's volume heat conduction elements is derived in Section 8.2, and the provisions for surface heat transfer are described in Section 8.3.

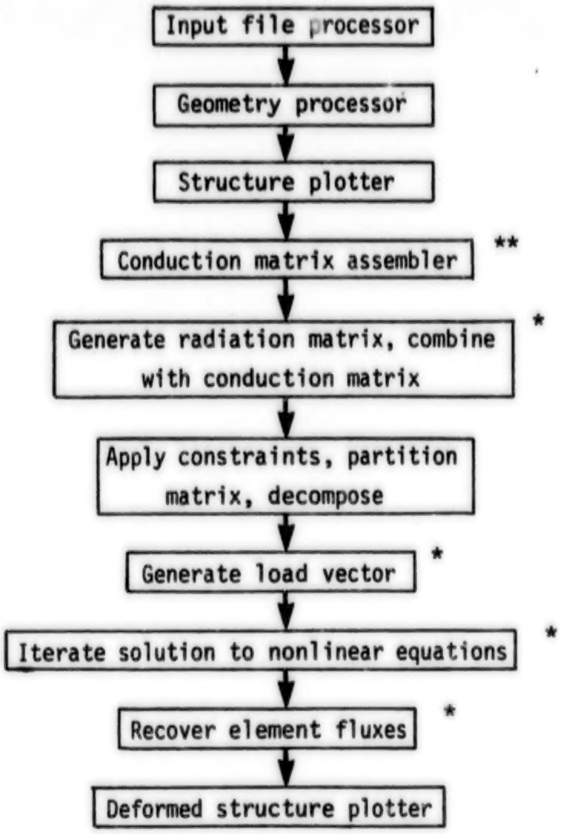
The components of the applied heat flow vector, {P} are associated either with surface heat transfer or with heat generated inside the volume heat conduction elements. The vector of nonlinear heat flow {N} results from surface radiation, from temperature-dependent surface convection, and from temperature-dependent heat conductivity.

In the case of linear static analysis, [B] and {N} are null, and Equation 1 is solved in the same manner as in linear static structural analysis. The flow diagram on Page 3.2-4 applies to this case. The user has the option to employ both single and multipoint constraints and many other specialized features normally associated with structural analysis. New solution techniques are used in nonlinear static analysis and in transient analysis. Flow diagrams for these cases are shown in Figures 1 and 2. Details of the computational procedures are explained in Section 8.4.

The output of a NASTRAN heat transfer analysis includes the temperature at gridpoints, the temperature gradients and heat fluxes within volume heat conduction elements, and the heat flow into surface elements. The heat flow into surface elements is further separated into components due to user-prescribed flux, radiation, and convective heat flux.

REFERENCE

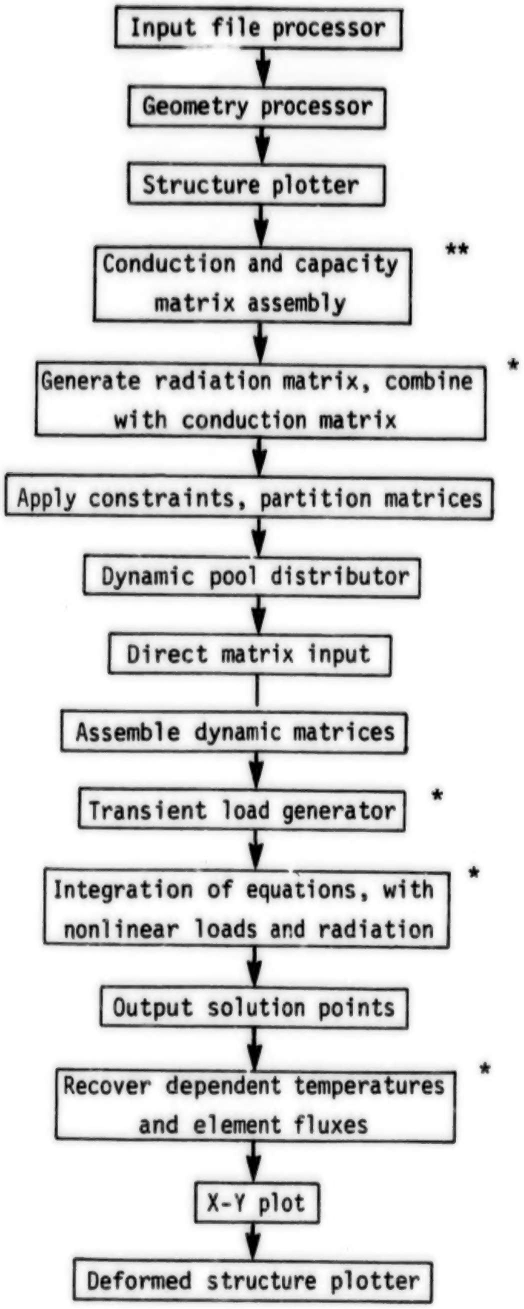
1. Zienkiewicz, O. C., and Cheung, Y. K., The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill Publishing Company Limited, London, 1967, Chapter 10.


Input file processor
Geometry processor
Structure plotter
Conduction matrix assembler **
Generate radiation matrix, combine with conduction matrix *
Apply constraints, partition matrix, decompose
Generate load vector *
Iterate solution to nonlinear equations *
Recover element fluxes *
Deformed structure plotter

* Modules used only for heat transfer analysis

** Existing modules to which heat transfer capability has been added.

Figure 1. Simplified flow diagram for thermal nonlinear statics analysis.


Input file processor
Geometry processor
Structure plotter
Conduction and capacity matrix assembly
**
Generate radiation matrix, combine with conduction matrix
*
Apply constraints, partition matrices
Dynamic pool distributor
Direct matrix input
Assemble dynamic matrices
Transient load generator
*
Integration of equations, with nonlinear loads and radiation
*
Output solution points
Recover dependent temperatures and element fluxes
*
X-Y plot
Deformed structure plotter

* Modules used only for heat transfer analysis

** Existing modules to which heat transfer capability has been added

Figure 2. Simplified flow diagram for thermal transient analysis.

8.2 VOLUME HEAT CONDUCTION ELEMENTS

The volume heat conduction elements are the same as NASTRAN structural elements. The elements for which heat conduction is available are listed in the following table:

Heat Conduction Elements	
Type	**Elements**
Linear	BAR, RØD, CØNRØD, TUBE
Planar	TRMEM, TRIA1, TRIA2, QDMEM, QUAD1, QUAD2
Solid of Revolution	TRIARG, TRAPRG
Solid	TETRA, WEDGE, HEXA1, HEXA2, IHEX1, IHEX2, IHEX3

Scalar elements, single point constraints, and multipoint constraints are also available for heat transfer analysis. The same connection and property cards are used for heat transfer and structural analysis. Linear elements have a constant cross-sectional area. For the planar elements, the heat conduction thickness is the membrane thickness. Elements with bending properties, such as BAR and TRIA1, have been included so that the user may use the same elements for the thermal and structural analyses of a given structure. The bending characteristics of the elements do not enter into heat conduction problems. The trapezoidal solid of revolution element, TRPRG, has been generalized to accept general quadrilateral rings (i.e., the top and bottom need not be perpendicular to the z-axis) for heat conduction only.

The heat conduction elements are composed of constant gradient lines, triangles and tetrahedra. The quadrilaterals are composed of overlapping triangles, and the wedges and hexahedra are formed from sub-tetrahedra in exactly the same way as for the structural case. The IHEXi elements are isoparametric hexahedron elements and are similar to the isoparametric solid elements described for structural analyses.

Thermal conductivity and capacity are specified on MAT4 (isotropic) and MAT5 (anisotropic) bulk data cards.

The heat conduction matrix for a volume heat conduction element may be derived from a thermal potential function in the same way that the stiffness matrix of a structural element is derived from the strain energy function. The thermal potential function is

PAGE 561 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

$$U = -\frac{1}{2} \int_V \tilde{q} \cdot \nabla u \, dV , \qquad (1)$$

where $\tilde{q}$ is the heat flux density, ∇u is the temperature gradient, and the integration is performed over the volume, V. The components of the heat flux are related to the components of temperature gradient by

$$q_i = - \sum_j k_{ij} \frac{\partial u}{\partial x_j} , \qquad (2)$$

where k_{ij} is an element of the material conductivity matrix and the index j is summed over the dimensions of the space (one, two, or three dimensions). Using Equation 2, Equation 1 may be expressed in matrix form as

$$U = \frac{1}{2} \int_V \left\lfloor \frac{\partial u}{\partial x_i} \right\rfloor [k_{ij}] \left\{ \frac{\partial u}{\partial x_j} \right\} dV . \qquad (3)$$

The temperature, u, at an interior point is a linear combination of the temperatures, $\{u_e\}$, at the vertices of the element, i.e.,

$$u = \lfloor L_e \rfloor \{u_e\} , \qquad (4)$$

where, in general, the elements of the row vector $\lfloor L_e \rfloor$ are functions of position. The thermal gradient vector is, therefore,

$$\left\{ \frac{\partial u}{\partial x_j} \right\} = [L_{e,j}]^T \{u_e\} , \qquad (5)$$

where the derivative matrix $[L_{e,j}]$ is, for the case of a two-dimensional triangular element,

$$[L_{e,j}] = \begin{bmatrix} \frac{\partial L_1}{\partial x} & \frac{\partial L_1}{\partial y} \\ \frac{\partial L_2}{\partial x} & \frac{\partial L_2}{y} \\ \frac{\partial L_3}{\partial x} & \frac{\partial L_3}{\partial y} \end{bmatrix} . \qquad (6)$$

In general the number of rows and columns of $[L_{e,j}]$ are the number of vertices of the element, and the dimension of the space, respectively. The substitution of Equation 5 into Equation 3 produces an expression with the form

$$U = \frac{1}{2} \lfloor u_e \rfloor [K^e] \{u_e\} \quad , \tag{7}$$

where the element heat conduction matrix is

$$[K^e] = \int_V [L_{e,i}][k_{ij}][L_{e,j}]^T dV \quad . \tag{8}$$

Equation 8 is a general form that is valid for all cases.

Elements of the heat capacity matrix [B] are calculated by the Lumped Mass method, see Section 5.5.

8.2.1 Constant Gradient Heat Conduction Elements

For the special case of a constant gradient element with homogeneous properties, $[L_{e,i}]$ and $[k_{ij}]$ in Equation 8 are constant within the element, so that

$$[K^e] = V_e[L_{e,i}][k_{ij}][L_{e,j}]^T \quad , \tag{9}$$

where V_e is the volume of the element. There is only one general type of constant gradient element for each type of space, i.e., a line segment for a one-dimensional space, a triangle for a two-dimensional space, and a tetrahedron for a three-dimensional space. In the constant gradient case, the elements of the vector $\{L_e\}$ are called the "natural coordinates" of the element. It is apparent from Equation 4 that each natural coordinate has unit value at one vertex and zero value at all other vertices of the element. The natural coordinates are obtained by the solution of

$$[H]\{L_e\} = \{f\} \quad , \tag{10}$$

where the specific forms for one, two and three dimensions are

one dimension
(line segment)

$$\begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ x \end{Bmatrix} \quad . \tag{11}$$

two dimensions
(triangle)

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} , \qquad (12)$$

three dimensions
(tetrahedron)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix} = \begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} . \qquad (13)$$

The determinant of the [H] matrix has a useful property, namely that:

for one dimension, $\det[H] = \ell$, the length of the segment,

for two dimensions, $\frac{1}{2}\det[H] = A$, the area of the triangle,

for three dimensions, $\frac{1}{6}\det[H] = V$, the volume of the tetrahedron.

In order to obtain the derivatives of $\{L_e\}$ required in Equation 9, we observe that, for the two-dimensional case,

$$\{L_e\} = [H]^{-1} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} , \qquad (14)$$

where $[H]^{-1}$ is a matrix of constant coefficients. The derivative matrix may, by comparing Equations 6 and 14, be expressed formally as

$$[L_{e,j}] = [H]^{-1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} , \qquad (15)$$

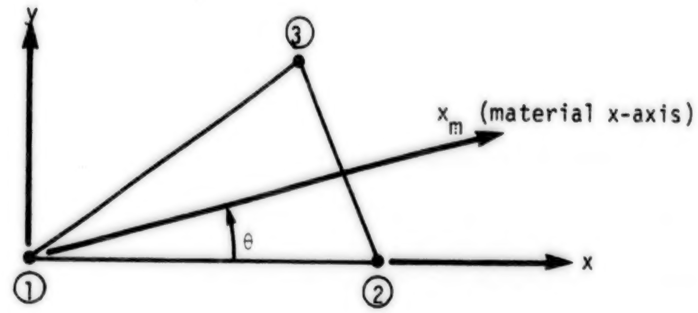
which means that $[L_{e,j}]$ is equal to the last two columns of $[H]^{-1}$. In general, for a space of (n) dimensions, $[L_{e,j}]$ is equal to the last (n) columns of $[H]^{-1}$.

For the case of the tetrahedron, the [H] matrix is inverted numerically, $[L_{e,j}]$ is taken to be the last three columns of $[H]^{-1}$, and $[K^e]$ is evaluated numerically from Equation 9. All calculations are performed in the basic coordinate system. For one and two-dimensional elements it is more practical to write explicit formulas for the natural coordinates. In fact, for one-dimensional elements the heat conduction matrix is simply

$$[K^e] = \frac{Ak}{\ell}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} , \tag{16}$$

where A is the cross-sectional area, k is the thermal conductivity, and ℓ is the length of the element.

In the case of a triangular element, the x-axis is taken along the side ① - ② as shown below:



The natural coordinates are, by inspection,

$$\left.\begin{aligned} L_1 &= 1 - \frac{x}{x_2} + \left(\frac{x_3}{x_2} - 1\right)\frac{y}{y_3} \\ L_2 &= \frac{x}{x_2} - \frac{x_3}{x_2}\frac{y}{y_3} \\ L_3 &= \frac{y}{y_3} \end{aligned}\right\} , \tag{17}$$

and the derivative matrix is

$$[L_{e,j}] = \left[\begin{array}{c|c} -1/x_2 & (x_3 - x_2)/x_2y_3 \\ \hline 1/x_2 & -x_3/x_2y_3 \\ \hline 0 & 1/y_3 \end{array}\right] \quad . \tag{18}$$

The material conductivity matrix $[k^m]$ is specified in the material coordinate system which makes an angle θ with respect to the element coordinate system as shown above. The conductivity matrix referred to the element coordinate system is

$$[k_{ij}] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} [k^m] \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{19}$$

Equations 18 and 19 are used in Equation 9 to obtain the heat conduction matrix for a triangle. The volume, V_e, is equal to the product of the surface area and the thickness.

For the triangular solid of revolution element (TRIARG) the differential volume to be used in Equation 8 is $2\pi r\ dr\ dz$, where r and z are cross-sectional coordinates. The temperature is assumed to be constant in the circumferential direction and to vary linearly over the cross-section. Thus, Equation 8 becomes

$$\begin{aligned} [K^e] &= [L_{e,i}][k_{ij}[L_{e,j}]^T\ 2\pi \int r\ dA \\ &= \frac{2\pi}{3}(r_1 + r_2 + r_3)A_e[L_{e,i}][k_{ij}][L_{e,j}]^T \quad , \end{aligned} \tag{20}$$

where A_e is the cross-sectional area. Equation 20 is identical to Equation 9 since the volume of a triangular ring is exactly

$$V_e = \frac{2\pi}{3}(r_1 + r_2 + r_3)A_e \quad . \tag{21}$$

Quadrilateral plane and revolution elements are formed by overlapping triangular elements in exactly the manner described in Section 5.8.3.1. Hexahedra and wedges are formed from subtetrahedra in exactly the manner described in Section 5.12.6.

Elements of the heat capacity matrix [B] are calculated by the Lumped Mass method, see Section 5.5. The total thermal capacity of an element is distributed equally among the connected grid points for lines, triangles and tetrahedra. In the case of triangular solid of revolution elements, the heat capacities lumped at the three grid circles are selected so that the total heat capacity and its center of gravity in the transverse plane are preserved. The equation for the heat capacity lumped at grid circle (i) is

$$b_i = \frac{\pi c_p A_e}{6} (r_i + r_1 + r_2 + r_3), \qquad i = 1, 2, 3, \tag{22}$$

where, in addition to previously defined terms, c_p is the heat capacity per unit volume.

The heat capacity matrices of elements formed by overlapping triangles or tetrahedra are computed by assigning one-half of the capacity to each overlapping set of sub-elements.

Thermal gradients are produced as part of the output, using Equation 5 and the various expressions derived above for the derivative matrix, $[L_{e,j}]$. The components of the heat flux are also output, using Equation 2, and the thermal gradient vector.

The temperature gradient and the heat flux are, of course, assumed constant over each subelement. In the case of overlapping elements, a weighted average is computed. The areas of the subtriangles are used as weighting functions in the case of planar elements, and the volumes of the subelements are used as weighting functions in the case of solids.

8.2.2 Three-Dimensional Isoparametric Solid Element Heat Transfer Matrices

The heat transfer conduction matrix for the three-dimensional isoparametric elements (References 1, 2 & 3) are derived by using Equation 8. For these three elements, the temperature, {u} at an exterior point is given by

$$\{u\} = [L_e]\{u_e\} \quad , \tag{23}$$

where $[L_e]$ is a function of the curvilinear coordinates ξ, η and ζ and is identical to the [N] matrix described in Section 5.13. The derivatives of $[L_e]$ with respect to the basic Cartesian coordinates x, y, z are calculated in exactly the same manner as presented in Section 5.13 for these elements, i.e.,

$$\begin{Bmatrix} \dfrac{\partial L_i}{\partial x} \\ \dfrac{\partial L_i}{\partial y} \\ \dfrac{\partial L_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \dfrac{\partial L_i}{\partial \xi} \\ \dfrac{\partial L_i}{\partial \eta} \\ \dfrac{\partial L_i}{\partial \zeta} \end{Bmatrix} \quad , \tag{24}$$

where [J] is the Jacobian Matrix, and the derivatives of L_i with respect to ξ, η and ζ are listed in Tables 5.13-1 and 5.13-2. Since, in general, the matrix $[L_{e,i}]$ is a function of ξ, η and ζ, the integration of Equation 8 is carried out by Gaussian Quadrature numerical integration, and the heat conduction matrix is calculated from the equation

$$[K^e] = \sum_{i=1} \sum_{j=1} \sum_{k=1} H_i \; H_j \; H_k([L_{e,i}][k_{ij}][L_{e,j}]^T)|J| \quad , \tag{25}$$

where

$$[L_{e,j}] = \begin{bmatrix} \dfrac{\partial L_1}{\partial x} & \dfrac{\partial L_1}{\partial y} & \dfrac{\partial L_1}{\partial z} \\ \dfrac{\partial L_2}{\partial x} & \dfrac{\partial L_2}{\partial y} & \dfrac{\partial L_2}{\partial z} \\ \vdots & \vdots & \vdots \\ \dfrac{\partial L_{NGP}}{\partial x} & \dfrac{\partial L_{NGP}}{\partial y} & \dfrac{\partial L_{NGP}}{\partial z} \end{bmatrix} \quad , \tag{26}$$

and NGP is the number of grid points in the element. NGP equals eight, twenty and thirty-two for the linear, quadratic and cubic isoparametric elements, respectively. The weight coefficients H_ℓ

and abscissa S_ℓ are given in Table 1.

Elements of the heat capacity matrix [B] are calculated by the coupled mass method (see Section 5.13.4). The equation for this matrix is

$$[B] = \int_V [L_e]^T [L_e] c_p \, dV \quad , \tag{27}$$

where c_p is the heat capacity per unit volume. Examination of this equation shows that [B] is identical to the structural mass matrix with the single exception that the heat capacity, c_p, is used instead of the material density. The heat capacity matrix is also evaluated using the Gaussian Quadrature formula

$$[B] = \sum_{i=1} \sum_{j=1} \sum_{k=1} H_i \; H_j \; H_k ([L_e]^T [L_e]) c_p |J| \quad . \tag{28}$$

HEAT TRANSFER ANALYSIS

REFERENCES

1. Zienkiewicz, O. C., and Cheung, Y. K., The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill Publishing Company Limited, London, 1967.

2. Zienkiewicz, O. C., and Parekh, C. J., "Transient Field Problems: Two-Dimensional and Three-Dimensional Analysis by Isoparametric Finite Elements," International Journal of Numerical Methods in Engineering, January-March, 1970.

3. Zienkiewicz, O. C., and Cheung, Y. K., "Finite Elements in the Solution of Field Problems," The Engineer, September 24, 1965, pp. 507-510.

8.3 SURFACE HEAT TRANSFER

Four types of surface heat transfer are provided for both steady state and transient analysis. The types are a prescribed heat flux, a convective heat flux due to the difference between the surface temperature and the local ambient temperature, radiation heat exchange, and a prescribed directed vector heat flux from a distant radiating source. In all cases the heat flux is applied to a surface element defined by gridpoints. There are six distinct types of surface elements:

1. POINT, a flat disc defined by a single gridpoint.
2. LINE, a rectangle defined by two gridpoints.
3. REV, a conical frustrum defined by two grid circles.
4. AREA3, a triangle.
5. AREA4, a quadrilateral.
6. ELCYL, an elliptic cylinder defined by two gridpoints. Its use is restricted to prescribed vector heat flux.

The user supplies the area, A, for POINT, and supplies a width, w, for LINE to be used with the distance, ℓ, between its gridpoints in calculating the surface area. For ELCYL the user specifies the principal radii of the cross-section. The surface area is calculated automatically in the other cases.

8.3.1 Prescribed Heat Flux

The user defines a distributed heat flux, Q, and the program calculates the vector of heat flows $\{P^e\}$ to be applied to the grid points connected to an element. The general form of the calculation for the jth component of $\{P^e\}$ is

$$P_j^e = A_j^e \, Q_j^e \quad , \tag{1}$$

where A_j^e is a subarea of the element associated with its jth vertex and Q_j^e is the heat flux at the jth vertex. There are two options for assigning heat fluxes to elements. In the first option (QBDY1 card) the user specifies a heat flux that is constant over the surface of the element. In option 2 (QBDY2 card) the user specifies separate heat fluxes at the vertices of the element, which are then used directly in Equation 1. In transient analysis, the time dependence of the flux is specified on a TLOAD card. The subareas A_j^e are calculated in the same manner as heat capacities. Thus, for LINE, A_j^e is one-half of the width multiplied by the distance between the end points, and for AREA3, A_j^e is equal to one-third of the total area. For AREA4, A_j^e is computed from the areas of the overlapping subtriangles connected to the jth grid point. For REV the total area is distributed to the two end points so as to preserve its center of gravity. ELCYL is not available for prescribed heat flux.

8.3.2 Convective Heat Flux

Convective heat flow into an element's grid points is described by the general relationship

$$\{P^e\} = [K^s]\{T_a - u_e\} \quad , \tag{2}$$

where $\{T_a - u_e\}$ is the difference between ambient and surface temperatures at the vertices of the element. The surface conduction matrix $[K^s]$ is calculated as follows for each surface element type. In the equations, K is the thermal conduction coefficient, which may be a function of temperature.

POINT:
$$K^s = KA \quad . \tag{3}$$

LINE:
$$[K^s] = \frac{Kw\ell}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad . \tag{4}$$

REV:
$$[K^s] = \frac{\pi K\ell}{6}\left[\begin{array}{c|c} 3r_1 + r_2 & r_1 + r_2 \\ \hline r_1 + r_2 & r_1 + 3r_2 \end{array}\right] \quad . \tag{5}$$

AREA3:
$$[K^s] = \frac{KA}{12}\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad , \tag{6}$$

where A is the area of the triangle.

AREA4:
$$K_{ij}^s = \frac{K}{24}\left[(1 + \delta_{ij})(a_1 + a_2 + a_3 + a_4) - (a_i + a_j)\right] \quad , \tag{7}$$

where $\delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$

a_i = area of the subtriangle which does not touch vertex (i).

The surface conduction matrices are, in each case, derived under the assumption that the temperature difference varies linearly over the surface of the element, except that, in the case of the quadrilateral (AREA4), the temperature difference is assumed to vary linearly over the surface of each overlapping subtriangle. Each ambient temperature, T_a, is assigned a degree of freedom in the analysis. The value of T_a may be specified on an SPC card in linear static analysis. The method used in nonlinear static analysis is described in Section 8.4.1. In transient thermal analysis, the various techniques for prescribing a displacement in transient structural analysis are used.

8.3.3 Radiation from a Distant Source

Radiation from a distant source, such as the sun, can be treated as a prescribed heat flux. The flux into a surface element depends upon the orientation of the radiation vector relative to the element. The total heat to a single element from a single distant source is given by

$$P = -aA(\bar{e}\cdot\bar{n})^{*}Q_0 \quad , \tag{8}$$

where P = power into the surface element from the source

Q_0 = power per unit area in the beam

A = surface area of the element

$\bar{e}$ = unit vector of radiation beam (the source is so distant that rays are parallel)

$\bar{n}$ = outward normal to surface

a = absorptivity (if $a < 1$, it is assumed that the reflected radiation is lost from the system)

$(\bar{e}\cdot\bar{n})^{*}$ is replaced by zero in the equation when $\bar{e}\cdot\bar{n}$ is positive, i.e., when the radiation comes from behind the surface.

No provision is made for shading by other surface elements.

In addition to the POINT, LINE and AREA elements, the elliptic cylinder element, ELCYL, can receive prescribed vector radiation, as shown below. An integration of the normal component of flux over the surface is needed to compute the power.

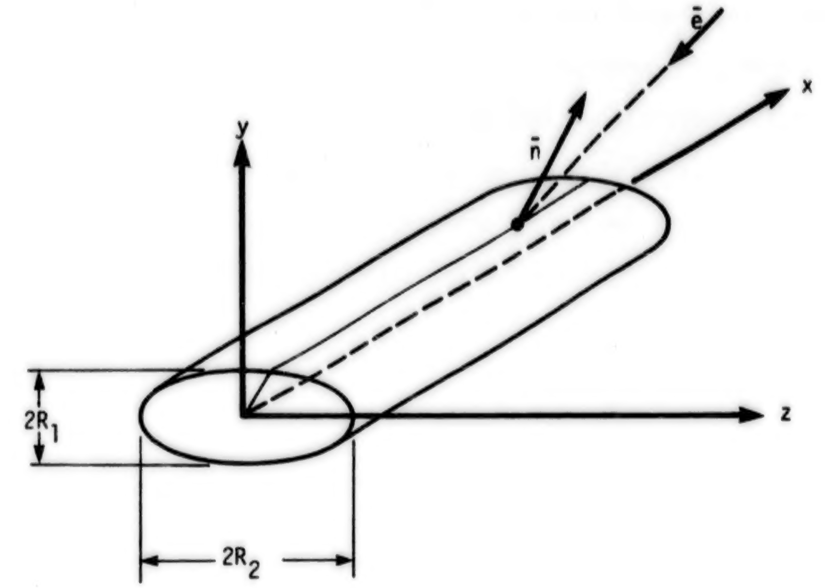


Figure 1. Elliptic Cylinder, ELCYL.

The result of the integration is

$$P = 2Q_0\ell\left[(e_yn_y)^2R_2^2 + (e_zn_z)^2R_1^2\right]^{1/2} \quad , \tag{9}$$

where e_y, e_z are components of $\bar{e}$; n_y, n_z are components of $\bar{n}$; and ℓ is the length of the cylinder.

In dynamic analysis the flux in the incident beam, Q_0, and the components of $\bar{e}$ may be prescribed functions of time. The latter provision is use[illegible] analysis of rotating spacecraft.

8.3.4 Radiation Exchange Between Surfaces

The relationship between the vector of radiative heat flows, $\{Q_g\}$, into grid points, and the grid point temperatures, $\{u_g\}$, is of the form (see, for instance, Reference 1)

$$\{Q_g\} = -[R_g]\{u_g + T_a\}^4 \quad . \tag{10}^{\dagger}$$

---

† If $\{T\}$ is a vector, $\{T\}^4$ is defined as the vector whose components are the fourth power of the elements of $\{T\}$.

The addition of T_a converts u_g to an absolute temperature scale. The grid point radiation matrix $[R_g]$ will be derived. As an intermediate step, an element radiation matrix, $[R_e]$, will be found which relates the radiative heat flow (power) into a finite element to the temperature of the element by

$$\{Q_e\} = -[R_e]\{U_e\}^4 \quad , \tag{11}$$

where U_e is the temperature of the element measured on an absolute scale and assumed to be constant over the element.

The radiation power into a surface area may be considered as the result of two effects: q_e^{out} (radiosity), the power per unit area leaving the surface, due either to direct radiation or to reflection of incoming radiation; and q_e^{in} (irradiation), the power per unit area arriving, due to the radiosity of the other elements. The radiation exchange formula is

$$[A]\{q\}_e^{in} = [F]\{q\}_e^{out} \quad , \tag{12}$$

where [A] is a diagonal matrix of areas, and [F] is a matrix of exchange coefficients whose elements are given by

$$F_{ij} = \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r_{ij}^2} \, dA_i \, dA_j \quad , \tag{13}$$

where r_{ij} is the length of a line connecting two points on the surfaces, and θ_i and θ_j are the angles between the connecting line and the normals to the surfaces. The range of integration must be limited to regions of the surfaces which "see" each other. The [F] matrix has units of area and is symmetric. Its elements are related to the more commonly used form factors (or shape factors) f_{ij} by

$$F_{ij} = A_j \, f_{ij} \quad , \tag{14}$$

where f_{ij} is the fraction of the power leaving element j which reaches element i. The elements of [F] are supplied by the user; it is expected that, in many cases, they will be computed by a separate computer program. The surface condition is

$$\{q\}_e^{out} = \sigma[E_\varepsilon]\{U_e\}^4 + [I - E_\alpha]\{q\}_e^{in} \quad , \tag{15}$$

where E_ε and E_α are diagonal matrices of emissivities and absorptivities and σ is the Stefan-Boltzmann constant. For gray bodies with no transmission, E_ε is equal to E_α, according to Kirchoff's Law.

The simultaneous solution of Equations 12 and 15 yields

$$\{q\}_e^{in} = \sigma[(A - F(I - E_\alpha))^{-1} F E_\varepsilon]\{U_e\}^4 \quad , \tag{16}$$

$$\{q\}_e^{out} = \sigma[E_\varepsilon + (I - E_\alpha)(A - F(I - E_\alpha))^{-1} F E_\varepsilon]\{U_e\}^4 \quad . \tag{17}$$

The net power exchange is

$$\{Q_e\} = [A]\left(\{q\}_e^{in} - \{q\}_e^{out}\right) \quad . \tag{18}$$

The element radiation matrix, found by substituting Equations 16 and 17 into Equation 18 and comparing the result with Equation 11, is

$$[R_e] = \sigma[AE_\varepsilon - AE_\alpha(A - F(I - E_\alpha))^{-1} F E_\varepsilon] \quad . \tag{19}$$

This matrix is symmetric if $E_\alpha = E_\varepsilon$ and F is symmetric. The transformation from element heat flow to grid point heat flow is given by

$$\{Q_g\} = [G]^T\{Q_e\} \quad , \tag{20}$$

where $[G]^T$ is a matrix of constant coefficients. The nonzero elements of $[G]^T$ are easily found for each element type. They are, in fact, the fractions of the area of the element attributed to the connected grid points, see Section 8.3.1. The same matrix transposed is used to interpolate the fourth powers of temperatures, i.e.,

$$\{U_e\}^4 = [G]\{u_g + T_a\}^4 \quad . \tag{21}$$

Combining Equations 20 and 21 with Equation 11, and comparing with Equation 10, it is seen that

$$[R_g] = [G]^T[R_e][G] \quad , \tag{22}$$

which is also a symmetric matrix if $[R_e]$ is symmetric. The net heat flow into the element due to radiation, which is available as output from NASTRAN, is

$$\{Q_e\} = -[R_e][G]\{u_g + T_a\}^4 \quad . \tag{23}$$

The sources of the information required to simulate radiation heat exchange are:

a. Parameters σ, T_a	User Supplied
b. Properties E_ε, E_α	
c. Exchange Matrix [F]	
d. Areas A	Computed from grid geometry and element properties
e. Transformation [G]	

HEAT TRANSFER ANALYSIS

REFERENCE

1. Jacob, M., Heat Transfer, Vol. II, John Wiley and Sons, 1957, pp. 1-24.

8.4 METHODS OF SOLUTION

The types of heat transfer problems that are solved with NASTRAN are described in Section 8.1. There are three types: linear steady-state analysis, nonlinear steady-state analysis, and transient analysis. Nonlinear relationships are optional in transient analysis.

Linear steady-state analysis uses the NASTRAN statics rigid format (format No. 1). The principal additions are subroutines for conduction and heat capacity matrix generation, described in Section 8.2, and for surface heat transfer, described in Section 8.3.

Flow diagrams for nonlinear steady-state analysis and for transient analysis are shown in Figures 1 and 2 of Section 8.1. Special features of the solutions are described in the subsections that follow.

8.4.1 Nonlinear Steady-State Analysis

The nonlinear properties permitted in steady-state heat transfer analysis with NASTRAN are radiation, temperature dependent film conduction, and temperature dependent volume heat conduction. The general form of the equation to be solved is

$$[K_{gg}]\{u_g\} + [R_{gg}]\{u_g + T_a\}^4 = \{q_g\} + \{P_g\} \quad . \qquad (1)^*$$

The temperature set $\{u_g\}$ includes degrees of freedom that are restrained by single point and multi-point constraints, see Section 3.3. The vector $\{q_g\}$ represents the forces of constraint. Sections 8.2 and 8.3 describe the manner in which the heat conduction matrix, $[K_{gg}]$, the radiation matrix, $[R_{gg}]$, and the applied heat flow vector, $\{P_g\}$, are formed from the properties of volume elements and surface elements. The elements of $[K_{gg}]$ may be functions of temperature.

The first step in the solution is to rewrite Equation 1 in terms of the set of temperatures, $\{u_n\}$, from which multi-point constraints have been removed. The procedures used are identical to those described in Section 3.5 for structural analysis. In order to avoid difficulties in interpolating temperatures to form the nonlinear terms, a restriction is placed on the form of the multi-point constraint relationships, namely that, if a grid point is adjacent to a volume or surface element with nonlinear properties, the constraint relationship is restricted to be an

---

*If $\{T\}$ is a vector, $\{T\}^4$ is defined as the vector whose components are the fourth power of the elements of $\{T\}$.

"equivalence." The term "equivalence" means that the constrained temperature is equal to one of the independent temperatures.

The form of the thermal equilibrium equation after the multi-point dependent temperatures $\{u_m\}$ have been eliminated is

$$[K_{nn}]\{u_n\} + [R_{nn}]\{u_n + T_a\}^4 = \{q_n\} + \{P_n\} \quad . \tag{2}$$

If $\{u_n\}$ is partitioned into $\{u_f\}$ (free points) and $\{u_s\}$ (single point constraints), Equation 2 may be written in partitioned form

$$\left[\begin{array}{c|c} K_{ff} & K_{fs} \\ \hline K_{sf} & K_{ss} \end{array}\right]\left\{\begin{array}{c} u_f \\ \hline u_s \end{array}\right\} + \left[\begin{array}{c|c} R_{ff} & R_{fs} \\ \hline R_{sf} & R_{ss} \end{array}\right]\left\{\begin{array}{c} u_f + T_a \\ \hline u_s + T_a \end{array}\right\}^4 = \left\{\begin{array}{c} 0 \\ \hline q_s \end{array}\right\} + \left\{\begin{array}{c} P_f \\ \hline P_s \end{array}\right\} \quad . \tag{3}$$

The components of $\{u_s\}$ have values prescribed by the user, and the lower half of Equation 3 is used to evaluate the single point "forces" of constraint $\{q_s\}$ during data recovery. Rearranging the top half of Equation 3 we obtain

$$[K_{ff}]\{u_f\} + [R_{ff}]\{u_f + T_a\}^4 = \{P_f\} - [K_{fs}]\{u_s\} - [R_{fs}]\{u_s + T_a\}^4 \quad . \tag{4}$$

Equation 4 is solved by an iterative method. The technique used is to expand $\{u_f\}$ into constant, linear, and higher order terms with respect to an initial estimate, $\{u_f^1\}$, supplied by the user. The linear terms are kept on the left hand side of Equation 4 and all other terms are placed on the right hand side, where they are evaluated precisely for the current estimate of $\{u_f\}$. If we define $\{L\}$ to be the left hand side of Equation 4, then the new left hand side is

$$\{L^*\} = \left[\frac{\partial L}{\partial u_f}\right]\{u_f\} = [K_{ff}^1]\{u_f\} + 4[R_{ff}]\lceil u_f^1 + T_a \rfloor^3\{u_f\} \quad , \tag{5}$$

where the partial derivatives are evaluated for $\{u_f\} = \{u_f^1\}$. Using this expression, Equation 4 may be written as

$$\begin{aligned}[K_{ff}^*]\{u_f\} = {} & [K_{ff}^1 - K_{ff}]\{u_f\} + [R_{ff}](4\lceil u_f^1 + T_a \rfloor^3\{u_f\} - \{u_f + T_a\}^4) \\ & + \{P_f\} - [K_{fs}]\{u_s\} - [R_{fs}]\{u_s + T_a\}^4 \quad ,\end{aligned} \tag{6}$$

where

$$[K^{*}_{ff}] = [K^{1}_{ff}] + 4[R_{ff}]\lceil u^{1}_{f} + T_a \rfloor^{3} \quad . \tag{7}$$

It is convenient, for computational purposes, to combine terms proportional to $\{u_f\}$ and $\{u_s\}$ on the right hand side of Equation 6 to produce terms proportional to $\{u_n\}$. Thus, if we define

$$\left.\begin{aligned}
\{u^{1}_{n}\} &= \begin{Bmatrix} u^{1}_{f} \\ -- \\ u_s \end{Bmatrix} \\
[K^{*}_{fs}] &= [K^{1}_{fs}] + 4[R_{fs}]\lceil u_s + T_a \rfloor^{3} \\
[K_{fn}] &= [K_{ff} \mid K_{fs}] \\
[K^{1}_{fn}] &= [K^{1}_{ff} \mid K^{1}_{fs}] \\
[R_{fn}] &= [R_{ff} \mid R_{fs}]
\end{aligned}\right\} \quad , \tag{8}$$

then Equation 6 may be written as

$$[K^{*}_{ff}]\{u_f\} = \{N_f\} \quad , \tag{9}$$

where

$$\{N_f\} = \{P_f - K^{*}_{fs}u_s\} - [K_{fn} - K^{1}_{fn}]\{u_n\} - [R_{fn}](\{u_n + T_a\}^{4} - 4\lceil u^{1}_{n} + T_a \rfloor^{3}\{u_n\}) \quad . \tag{10}$$

The first term in Equation 10 is a constant, and the other terms are functions of temperature. Equation 9 is an exact relationship. The iteration algorithm consists of evaluating $\{N_f\}$ for $\{u_n\} = \{u^{i}_{n}\}$, the current estimate of the temperature distribution, and of solving Equation 9 to obtain a new estimate, $\{u^{i+1}_{f}\}$, of the unknown temperatures. The starting vector is $\{u^{1}_{f}\}$, supplied by the user.

The algorithm is simple enough, but the number of iterations to obtain satisfactory convergence (if indeed convergence can be achieved) remains an open question. The question of convergence can be treated without difficulty in a small neighborhood of the correct solution within which the nonlinear load may be approximated as a linear function of the error in the temperature distribution. The iteration algorithm is

$$[K^*_{ff}]\{u_f^{i+1}\} = \{N_f^i\} \quad . \tag{11}$$

As an approximation, let

$$\{N_f^i\} = \{N_f\} + [C]\{u_f^i - u_f\} = [N_f] + [C]\{\delta u^i\} \quad , \tag{12}$$

where [C] is the matrix of the partial derivatives of $\{N_f\}$ with respect to changes in $\{u_f\}$. Substituting Equation 12 into Equation 11 and using Equation 9, the iteration algorithm is, approximately,

$$[K^*_{ff}]\{\delta u^{i+1}\} = [C]\{\delta u^i\} \quad . \tag{13}$$

Equation 13 resembles the power method of eigenvalue extraction, see Section 10.4, and its convergence is related to the distribution of the eigenvalues of the associated eigenvalue problem,

$$[K^*_{ff} \; - \lambda C]\{\delta u\} = 0 \quad . \tag{14}$$

In order to establish the condition for convergence, expand the iterates $\{\delta u^i\}$ and $\{\delta u^{i+1}\}$ in terms of the eigenvectors $\{\phi_j\}$, i.e.,

$$\{\delta u^i\} = \sum_j \alpha_j^i\{\phi_j\} \quad , \tag{15}$$

$$\{\delta u^{i-1}\} = \sum_j \alpha_j^{i-1}\{\phi_j\} \quad . \tag{16}$$

It can be proved quite generally (see, for example, Section 10.4.4.3) that a property of the linearized iteration algorithm is that

$$\alpha_j^{i-1} = \lambda_j \; \alpha_j^i \quad , \tag{17}$$

where λ_j is the eigenvalue corresponding to $\{\phi_j\}$. Thus, it is seen that, if $|\lambda_j| < 1$, α_j^i will increase in magnitude at each iteration and the algorithm will be divergent. The necessary and sufficient condition for convergence in a small neighborhood of the correct solution is that all eigenvalues of Equation 14 have magnitudes greater than one.

NASTRAN provides both an estimate of the lowest eigenvalue and an estimate of the error in the solution after each iteration. If the iteration has proceeded to the point where one eigenvector, $\{\phi_1\}$, dominates the solution, it is seen from Equations 15, 16, and 17 that

$$\{\delta u^i - \delta u^{i-1}\} = \frac{1}{\lambda_1}\{\delta u^{i-1} - \delta u^{i-2}\} \quad , \tag{18}$$

so that the ratios of successive increments in the elements of the solution vector provide an estimate of the lowest eigenvalue. By analogy with a procedure used in the inverse power method with shifts, see Section 10.4, a single weighted estimate is obtained by multiplying both sides of Equation 18 by the transpose of the nonlinear load vector. Thus, the estimate is

$$\lambda_1^i = \frac{\{N_f^i\}^T\{u_f^{i-1} - u_f^{i-2}\}}{\{N_f^i\}^T\{u_f^i - u_f^{i-1}\}} \quad . \tag{19}$$

Equation 19 is evaluated after every iteration starting with the third, i = 4.

The vector $\{\delta u^i\}$ is the error in the solution at the beginning of the ith iteration. In order to obtain a measure of the error, we observe, from Equations 15, 16 and 17, that if only one eigenvector is present

$$\{\delta u^i - \delta u^{i-1}\} = (1 - \lambda_1)\{\delta u^i\} \quad . \tag{20}$$

The measure of the error in temperature used in NASTRAN is the ratio of the work done by the nonlinear loads acting on the error vector to the work done by the nonlinear loads acting on the total solution, i.e.,

$$\epsilon_T^i = \left|\frac{\{N^i\}^T\{\delta u^i\}}{\{N^i\}^T\{u_f\}}\right| = \left|\frac{1}{(\lambda_1^i - 1)}\,\frac{\{N_f^i\}^T\{u_f^i - u_f^{i-1}\}}{\{N_f^i\}^T\{u_f^i\}}\right| \quad . \tag{21}$$

Another error measure is also provided, which measures the error in the applied heat flux, including nonlinear terms. That measure is

$$\epsilon_p^i = \frac{||N_f^i - [K_{ff}^*]u_f^i||}{||N_f^1||} = \frac{||N_f^i - N_f^{i-1}||}{||N_f^1||} \quad , \tag{22}$$

where $||X||$ is the Euclidean norm of the vector $\{X\}$.

The iteration algorithm will terminate for any of the following reasons:

a) ε_T^i is less than a user-specified value and also $|\lambda_1^i| > 1$: Normal convergence.

b) $|\lambda_1^i| < 1$ for $i \geq 4$: The algorithm is assumed to be divergent.

c) The number of iterations reaches the maximum number specified by the user.

d) The available time is used up.

In all cases, the values of ε_T^i, ε_p^i and λ_1^i may be output for every iteration, and the solution vector for the last iteration will be output.

Radiated heat flux is proportional to the fourth power of the temperature, thereby providing a very strong nonlinear effect if the radiation terms are large compared to other terms. In order to guage the effect of radiation on convergence of the iteration algorithm, consider an isolated perfectly-conducting body in thermal equilibrium with radiation from distant sources. The thermal equilibrium equation is

$$Ru^4 = P \quad , \tag{23}$$

where u is measured on an absolute scale, and P is constant. The user supplies an estimate of the temperature, u_1. The iteration algorithm used by NASTRAN is, in accordance with preceding discussion,

$$(4Ru_1^3)u_{i+1} = P - R[(u_i)^4 - 4u_1^3u_i] \quad . \tag{24}$$

The derivative of the right hand side at the correct solution ($u_i = u$) is

$$C = -4R(u_1^3 - u^3) \quad , \tag{25}$$

so that the eigenvalue problem corresponding to Equation 14 is

$$[4Ru_1^3 - \lambda 4R(u_1^3 - u^3)]\delta u = 0 \quad . \tag{26}$$

The eigenvalue is

$$\lambda = \frac{u_1^3}{u_1^3 - u^3} \quad . \tag{27}$$

The critical value for convergence, $\lambda = -1$, is achieved if

$$u_1^3 = \frac{1}{2} u^3 \quad ,$$

or

$$u_1 = .794u \ ,$$

Thus, the solution converges if u_1 is greater than about 80% of the correct temperature, measured on an absolute scale. The user can assure convergence, at the expense of extra iterations, by overestimating the temperature.

8.4.2 Transient Analysis

The nonlinear terms permitted in transient heat transfer analysis include radiation and the general purpose nonlinear elements described in Section 11.2. Nonlinear heat conduction and heat capacity are not permitted. The reason is that the computational effort required to recalculate the heat conduction and heat capacity matrices at each time step by the finite element method used in NASTRAN is judged to be excessive. The general purpose nonlinear elements can, however, be used to represent nonlinear surface film conduction and other relatively simple nonlinear relationships.

The general equation solved in transient analysis has the form

$$[K]\{u\} + [B]\{\dot{u}\} = \{P\} + \{N\} \quad . \tag{1}$$

The conduction matrix includes linearized radiation terms. It is, in fact, identical to $[K_{ff}^*]$ given by Equation 7 of Section 8.4.1, except possibly for terms due to "extra points," see Section 9.3.2. The nonlinear term in Equation 1 is

$$\{N\} = \{N^e\} + [R](4\lceil u^1 + T_a \rfloor^3\{u\} - \{u + T_a\}^4) \quad , \tag{2}$$

where $\{N^e\}$ is due to general purpose nonlinear elements and the second term is due to radiation. An option is available to substitute $\{u^1\}$ for $\{u\}$ in the second term, which reduces it to a constant vector and which, thereby, linearizes the effect of radiation.

The load vector $\{P\}$ may be formed in the same manner as for static heat transfer analysis with certain parameters permitted to be functions of time. These include the prescribed volumetric and surface heat fluxes, and the prescribed vector heat flux. In the latter case, both the direction and the magnitude of the heat flux are permitted to be functions of time. The user also has available the methods used to prescribe transient loads in structural dynamic analysis, see Section 11.1. Prescribed temperatures at grid points, and the ambient temperatures used for film heat transfer are treated in the same manner as prescribed displacements in dynamic analysis. The user connects a large scalar conduction element, K_0, to the grid point in question and also applies a thermal load $P = TK_0$ to the grid point, where T is the desired temperature function of time.

The algorithm used to integrate Equation 1 has been selected with the following criteria in mind:

1. Unconditional stability for linear problems, regardless of the size of the time step,
2. Ability to handle a singular heat capacity matrix,
3. Good stability for nonlinear problems,
4. Good efficiency,
5. High accuracy.

A useful general observation is that stability, efficiency and accuracy are conflicting requirements that must be compromised. The algorithm that has been selected can satisfy the first two criteria and scores reasonably well on the last three. Basically, it is a difference equation approximation to Equation 1 with a free parameter that is adjusted to produce a compromise of the stability, efficiency and accuracy requirements. In this respect it is analogous to the Newmark β method used in structural dynamics, see Section 11.3. The form of the difference equation is

$$[K]\{\beta\, u_{n+1} + (1 - \beta)u_n\} + \frac{1}{\Delta t}[B]\{u_{n+1} - u_n\} =$$

$$\{\beta P_{n+1} + (1 - \beta)P_n\} + (1 + \beta)\{N_n\} - \beta\{N_{n-1}\} \quad . \qquad (3)$$

The subscript n refers to the nth time step. The parameter, β, may be selected by the user in the range $0 < \beta < 1$. Putting terms proportional to $\{u_{n+1}\}$ on the left side yields the iteration algorithm

$$\left[\frac{1}{\Delta t}B + \beta K\right]\{u_{n+1}\} = \left[\frac{1}{\Delta t}B - (1-\beta)K\right]\{u_n\}$$

$$+ \beta\{P_{n+1}\} + (1-\beta)\{P_n\} + (1+\beta)\{N_n\} - \beta\{N_{n-1}\} \quad . \tag{4}$$

The matrix $\left[\frac{1}{\Delta t}B + \beta K\right]$ is first decomposed into its triangular factors from which the equations are solved at each time step using a forward and backward substitution procedure, see Section 2.3. The time step, Δt, may be changed at discrete times by the user. Certain values of the parameter β result in well-known algorithms, viz.,

$\beta = 0$: Euler integration

$\beta = 1/2$: Central differences

$\beta = 1$: Backward differences

Euler integration (β=0) is usually the most efficient choice because only the [B] matrix, which is frequently diagonal, is decomposed. However, Euler integration cannot be used if [B] is singular and it suffers with respect to both stability and accuracy as will be seen.

The effect of β on stability will be examined for the linear case, for which the matrix equation of motion is

$$[K]\{u\} + [B]\{\dot{u}\} = P \quad . \tag{5}$$

A more convenient set of equations is obtained by a transformation of {u} into modal coordinates, $\{\xi_i\}$:

$$\{u\} = [\phi]\{\xi_i\} \quad , \tag{6}$$

where each column of $[\phi]$ is an eigenvector of Equation 5. The equation for each modal coordinate is uncoupled from the others and has the form

$$\dot{\xi}_i + \lambda_i\,\xi_i = P_i \quad , \tag{7}$$

where λ_i is the eigenvector and P_i is the generalized force on ξ_i. The system of equations is stable if all $\lambda_i \geq 0$.

Applying the integration algorithm to Equation 7 we obtain

$$\frac{1}{\Delta t}\left(\xi_{n+1} - \quad\right) + \lambda_i\left(\beta\,\xi_{n+1} + (1 - \beta)\xi_n\right) = \beta\,P_{n+1} + (1 - \beta)P_n \quad , \tag{8}$$

where the subscript (i) has been omitted for clarity. The solution for the homogeneous case ($P_n = P_{n+1} = 0$) has the property that

$$\xi_{n+1} = E\,\xi_n \quad , \tag{9}$$

where E is a constant, called the shift operator. If $|E| \leq 1$, the homogeneous solution is stable because it approaches zero for large n. By substituting Equation 9 into Equation 8 for the homogeneous case, we obtain

$$\left[\frac{1}{\Delta t}(E - 1) + \lambda_i(\beta\,E + 1 - \beta)\right]\xi_n = 0 \quad . \tag{10}$$

Setting the coefficient of ξ_n to zero, which must occur if ξ_n is not to be zero, produces a functional relationship between E, β, and $\lambda_i\Delta t$, which may be expressed in the form

$$\lambda_i\Delta t = \frac{1 - E}{E\,\beta + 1 - \beta} \quad . \tag{11}$$

The range of E for stability is $-1 \leq E \leq 1$. Substitution of the upper limit into Equation 11 is seen to produce no restriction on the time step. Substitution of the lower limit, however, gives as a stability limit

$$\lambda_i\Delta t = \frac{2}{1 - 2\beta} \quad . \tag{12}$$

Thus, if $\beta = 0$ (Euler integration) the stability limit is $\lambda_i\Delta t = 2$. Since λ_i is the reciprocal of the time constant of the ith mode of the system, the practical restriction on Euler integration is that the time step can be no greater than twice the smallest decay time constant of the system. If $\beta = 1/2$, there is seen to be no limit on the time step, nor is there for $\beta > 1/2$, which can most readily be seen by solving Equation 11 for E:

$$E = \frac{1 - (1 - \beta)\lambda_i\Delta t}{1 + \beta\,\lambda_i\Delta t} \quad . \tag{13}$$

From the viewpoint of stability then, β should be chosen greater than or equal to 1/2. For linear problems $\beta = 1/2$ is adequate, but for nonlinear problems in which the nonlinear terms must necessarily be evaluated at the nth and earlier time steps, a larger value of β may be advisable.

Insight into the question of accuracy can be gained by examining the eigenvalues produced by the integration algorithm and comparing them with the eigenvalues of the real system. The eigenvalue, Λ_i, produced by numerical integration is defined implicitly by

$$\xi_{n+1} = e^{-\Lambda_i \Delta t} \xi_n \quad , \qquad (14)$$

or, by comparison with Equation 9,

$$\Lambda_i = \frac{-1}{\Delta t} \ln E \quad , \qquad (15)$$

so that, using Equation 13,

$$\Lambda_i = \frac{-1}{\Delta t} \ln \left(\frac{1 - (1 - \beta)\lambda_i \Delta t}{1 + \beta \lambda_i \Delta t} \right) \quad . \qquad (16)$$

If $\lambda_i \Delta t$ is assumed to be less than one, Equation 16 can be evaluated by power series expansion with the result

$$\Lambda_i = \lambda_i \left[1 + (\beta - 1/2)(\lambda_i \Delta t) - \left(\frac{1}{12} + (\beta - 1/2)^2 \right) (\lambda_i \Delta t)^2 + --- \right] \quad . \qquad (17)$$

It is seen that, if the time step, Δt, is small compared to the decay time constant of the mode, $1/\lambda_i$, the error will be a minimum near $\beta = 1/2$. Since efficiency or stability considerations will be overriding in many cases, the choice of β is left to the user. The default value, in the event that the user declines to make a choice, is $\beta = 0.55$.

The provisions for initial conditions are as follows. The initial thermal load (for Equation 4 at $n = 0$) is taken as

$$\{P_0\} = [K]\{u_0\} - \{N_0\} \quad , \qquad (18)$$

which sets $\{\dot{u}\}$ to zero initially (see Equation 1). Since $\{u_n\}$ is not defined for negative n, the nonlinear load at $t = -\Delta t$ is taken to be

$$\{N_{-1}\} = \{N_0\} \quad . \qquad (19)$$

Equations 18 and 19 have the property that they yield smooth results when step loads are applied to degrees of freedom without thermal capacity. Special conditions are also needed if it is desired to change the time step. The situation is similar to the starting equations except that the new initial velocity vector, $\{\dot{u}\}$, is set equal to the old final vector. Let N be the index of the last step with the previous time step Δt_1. Let Δt_2 be the new time step and let the time step counter be reset to zero. The new initial conditions are

$$\{u_0\} = \{u_N\} \quad , \tag{20}$$

$$\{\dot{u}_0\} = \frac{1}{\Delta t_1}\{u_N - u_{N-1}\} \quad . \tag{21}$$

The new initial thermal load is

$$\{P_0\} = [K]\{u_0\} - \{N_0\} - [B]\{\dot{u}_0\} \quad . \tag{28}$$

Interpolation is used to produce an estimate of the nonlinear load at $t = -\Delta t_2$:

$$\{N_{-1}\} = \frac{\Delta t_2}{\Delta t_1}\{N_{N-1}\} + \left(1 - \frac{\Delta t_2}{\Delta t_1}\right)\{N_N\} \quad . \tag{23}$$

These provisions are designed to minimize discontinuities associated with time step changes. The coefficient matrices in Equation 4 are recomputed, and the matrix coefficient of $\{u_{n+1}\}$ is decomposed before continuing the integration with the new initial values.

9. ORGANIZATION OF DYNAMIC ANALYSIS

9.1 GENERAL PROBLEM FLOW IN DYNAMIC ANALYSIS

Figure 1 is a flow diagram showing the major functional modules employed in the solution of dynamic problems by the displacement method, except for aeroelastic problems which are described in Figure 1 and 2 of Section 17.6. Three basic types of analysis are performed (Eigenvalue Extraction, Frequency Response Analysis, and Transient Response Analysis) according to either of two methods of problem formulation (Direct or Modal). In all there are nine different paths through the flow diagram of Figure 1, corresponding to the following seven rigid formats for dynamic analysis, which supplement the six rigid formats for static analysis described in Section 3.2. (The numbers are the Rigid Format numbers assigned in the Program.)

3. Normal Modes Analysis
7. Direct Complex Eigenvalue Analysis
8. Direct Frequency and Random Response Analysis
9. Direct Transient Response Analysis
10. Modal Complex Eigenvalue Analysis
11. Modal Frequency and Random Response Analysis
12. Modal Transient Response Analysis
13. Normal Modes Analysis with Differential Stiffness
15. Normal Modes Analysis using Cyclic Symmetry

In the modal method of dynamic problem formulation, the vibration modes of the structure in a selected frequency range are used as degrees of freedom, thereby reducing the number of degrees of freedom while maintaining accuracy in the selected frequency range. In the direct method, the degrees of freedom are simply the displacements at grid points.

It is important to have both direct and modal methods of dynamic problem formulation in order to maximize efficiency in different situations. The modal method will usually be more efficient in problems where a small fraction of all of the modes are sufficient to produce the desired accuracy and where the bandwidth of the direct stiffness matrix is large. The direct method will usually be more efficient for problems in which the bandwidth of the direct stiffness matrix is small, and for problems with dynamic coupling in which a large fraction of the vibration modes are required to produce the desired accuracy. For problems without dynamic coupling, i.e., for problems in which the matrices of the modal formulation are diagonal, the modal method will frequently be more efficient, even though a large fraction of the modes are needed. The choice of method is of course, left to the user.

The flow diagram in Figure 1 is simplified to the extent that it shows only the major operations that are performed. Complete descriptions of the sequences of module calls for all rigid formats are contained in the NASTRAN User's Manual. The functions indicated in Figure 1 are described in succeeding subsections of the Theoretical Manual as follows.

Section 9.2 - Real Eigenvalue Analysis, READ
Complex Eigenvalue Analysis, CEAD

Section 9.3 - Dynamic Pool Distributor, DPD
Direct Dynamic Matrix Assembler, GKAD
Modal Dynamic Matrix Assembler, GKAM

Section 9.4 - Dynamic Data Recovery, DDR

Section 11. - Transient Response Analysis, TRD

Section 12. - Frequency Response Analysis, FRRD
Random Analysis Module, RANDØM

The use of the real eigenvalue analysis module, READ, for buckling problems has been described in Section 3.2. Section 9.2 contains information concerning the selection of methods, the checks that are performed, and other organizational details for both real and complex eigenvalue analysis. The mathematics of the eigenvalue extraction methods employed with NASTRAN are developed in Section 10.

A basic feature of NASTRAN is its generality and flexibility with regard to the specification of input data for dynamic analysis. The general means provided for specifying damping, control system characteristics, aerodynamic influence coefficients, etc., are described in Section 9.3. Further information on special problem formulation techniques for dynamic analysis is given in Section 14. Hydrodynamic theories are discussed in Section 16. Aerodynamic methods, using special rigid formats, are discussed in Section 17.

GENERAL PROBLEM FLOW IN DYNAMIC ANALYSIS

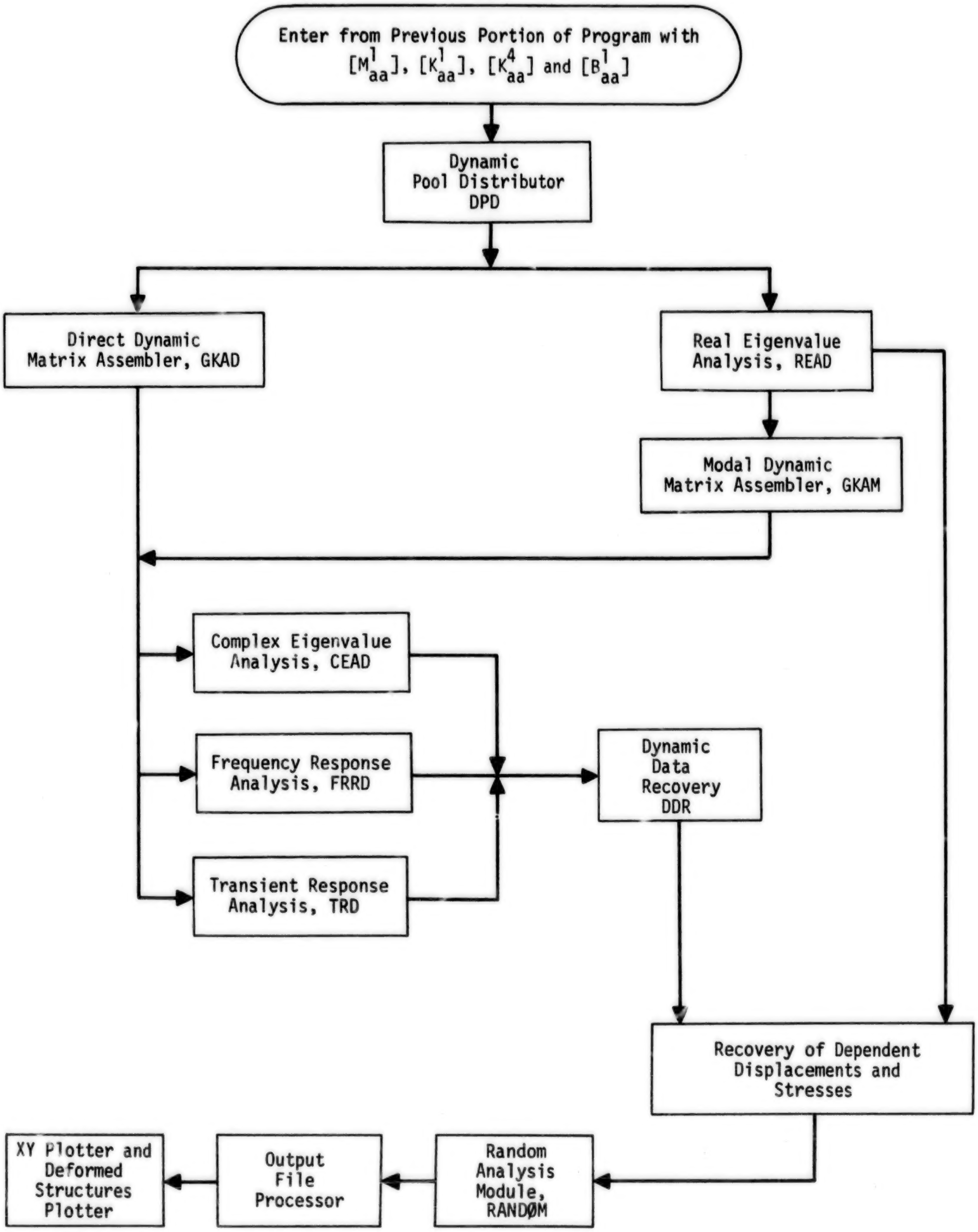

Enter from Previous Portion of Program with
$[M^1_{aa}]$, $[K^1_{aa}]$, $[K^4_{aa}]$ and $[B^1_{aa}]$
Dynamic Pool Distributor DPD
Direct Dynamic Matrix Assembler, GKAD
Real Eigenvalue Analysis, READ
Modal Dynamic Matrix Assembler, GKAM
Complex Eigenvalue Analysis, CEAD
Frequency Response Analysis, FRRD
Transient Response Analysis, TRD
Dynamic Data Recovery DDR
Recovery of Dependent Displacements and Stresses
Random Analysis Module, RANDØM
Output File Processor
XY Plotter and Deformed Structures Plotter

Figure 1. Simplified flow diagram for dynamic analysis by the displacement method.

9.2 EIGENVALUE ANALYSIS

Real and complex eigenvalue analyses are performed in NASTRAN by separate modules, as indicated in Figure 1 of the preceding section. The real eigenvalue analysis module, READ, is used to obtain structural vibration modes from the symmetric mass and stiffness matrices, $[M_{aa}]$ and $[K_{aa}]$, generated in the statics part of the program, and to obtain buckling modes from the elastic stiffness and differential stiffness matrices, $[K_{aa}]$ and $[K_{aa}^d]$. All other eigenvalue problems, i.e., those with matrix terms from additional sources, are solved in the complex eigenvalue analysis module, CEAD. Examples include the vibration modes of damped systems, the stability analysis (flutter) of structures with aerodynamic coupling and/or control system feedback, and the buckling of structures with nonsymmetric terms in the differential stiffness matrix.

The eigenvectors and eigenvalues produced by READ may be used to generate modal coordinates for further dynamic analysis by the modal method. The results of CEAD, on the other hand, are an end product. The organization of the real and complex eigenvalue modules is discussed below in separate subsections.

9.2.1 Real Eigenvalue Analysis

A flow diagram for the real eigenvalue analysis module, READ, is shown in Figure 1. The user has a choice of three methods for solving buckling problems and of four methods for solving vibration mode problems. More than one method is provided in order to optimize efficiency for different types of problems and also to provide back up in case one method gives unsatisfactory results in a particular case. The tridiagonal method is not provided as an option for buckling problems because it would require that the differential stiffness matrix be negative definite, which is seldom the case. The methods of eigenvalue extraction used in NASTRAN are described in detail in Section 10. The introduction, Section 10.1, includes a comparison of the methods for different types of problems.

The general form of the eigenvalue problem for vibration modes is

$$[K_{aa} - \lambda M_{aa}]\{u_a\} = 0 \quad . \qquad (1)$$

The eigenvalues $\lambda_i = \omega_i^2$ are the squares of the natural vibration frequencies. The results of the calculation performed by the module are the eigenvalues, λ_i, and corresponding eigenvectors $\{\phi_{ai}\}$, normalized in one of three optional ways. At the user's request the modal mass matrix

PAGE 595 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

$$[m_{ii}] = [\phi_{ai}]^T [M_{aa}][\phi_{ai}] \quad , \tag{2}$$

is calculated and checked for orthogonality of the modes (see page 9.2-5). The columns of $[\phi_{ai}]$ are the eigenvectors, $\{\phi_{ai}\}$.

The general form of the eigenvalue problem for buckling is

$$[K_{aa} + \lambda K_{aa}{}^{d}]\{u_a\} = 0 \quad . \tag{3}$$

The eigenvalues λ_i are the factors by which the static loading condition must be multiplied to produce buckling. The results of the calculation performed by the module are the eigenvalues, λ_i, and the corresponding eigenvectors $\{\phi_{ai}\}$. Additional data processing is performed by other modules, as indicated in Figure 1 of Section 9.1.

The user has a choice, for either type of problem, of the number of eigenvalues to be extracted and/or of the range of λ within which they are to be extracted. The available options, which vary slightly for the different methods, are explained in Section 10.

Rigid body vibration modes are calculated by a separate procedure provided that a set of reaction (support) points, u_r, have been specified by the user (see Section 3.5.5). This is done to improve efficiency and, in some cases, reliability. The supports are not applied to the structure during calculation of the remaining eigenvalues. If the user does not specify the reaction points (or if he specifies an insufficient number of them) the (remaining) rigid body modes will be calculated by the method selected for the finite frequency modes.

It will be recalled, Section 3.5.5, that the rigid body mass matrix referred to the reaction points, $[m_r]$, and the rigid body transformation matrix, $[D]$, in the equation relating the remaining degrees of freedom, u_ℓ, to u_r,

$$\{u_\ell\} = [D]\{u_r\} \ , \tag{4}$$

are computed in the static portion of the program. The rigid body modes are a set of vectors

$$\{\phi_{ao}\} = \left\{\begin{array}{c} \phi_{\ell o} \\ \hline \phi_{ro} \end{array}\right\} = \left\{\begin{array}{c} D\phi_{ro} \\ \hline \phi_{ro} \end{array}\right\} \quad , \tag{5}$$

such that the modal mass matrix,

597

$$[m_o] = [\phi_{ro}]^T [m_r][\phi_{ro}] \quad , \tag{6}$$

is diagonal and normalized. The Schmidt orthogonalization procedure (Reference 1) is used to obtain the vectors $\{\phi_{ro}\}$ that are the columns of the modal matrix $[\phi_{ro}]$. The specific procedure used in NASTRAN is as follows:

1. Define a set of vector delta functions

$$\{I_1\} = \begin{Bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} , \quad \{I_2\} = \begin{Bmatrix} 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} \quad , \text{ etc.,} \tag{7}$$

   a set of unnormalized eigenvectors $\{V_{io}\}$, $i = 1, 2, \ldots r$, and a set of normalized eigenvectors $\{\phi_{io}\}$, $i = 1, 2, \ldots r$.

2. The relationship between $\{\phi_{io}\}$ and $\{V_{io}\}$ is

$$\{\phi_{io}\} = \frac{\{V_{io}\}}{(\{V_{io}\}^T [m_r]\{V_{io}\})^{1/2}} \quad , \tag{8}$$

   i.e., $\{\phi_{io}\}$ is normalized to unit generalized mass.

3. Set the first unnormalized eigenvector equal to the first delta function.

$$\{V_{1o}\} = \{I_1\} \quad . \tag{9}$$

4. The second unnormalized eigenvector is obtained from

$$\{V_{2o}\} = \{I_2\} - \alpha_{12}\{\phi_{1o}\} \quad , \tag{10}$$

   where

$$\alpha_{12} = \{\phi_{1o}\}^T [m_r]\{I_2\} \quad . \tag{11}$$

5. Generalize, to obtain the remaining eigenvectors

$$\{V_{io}\} = \{I_i\} - \sum_{j=1}^{i-1} \alpha_{ji}\{\phi_{jo}\} \quad , \quad i = 2, 3, \ldots r, \tag{12}$$

where

$$\alpha_{ji} = \{\phi_{jo}\}^T [m_r]\{I_i\} \quad . \tag{13}$$

Turning to another peripheral matter, the form of the tridiagonal method used in NASTRAN requires that the eigenvalue problem be stated in the form

$$[J - \lambda I]\{w\} = 0 \quad , \tag{14}$$

where [I] is the identity matrix and [J] is symmetrical. The operations indicated in blocks 4 and 5 of Figure 1 put the problem in this form. The first operation is to perform a Cholesky decomposition of the matrix

$$[M_{aa}] = [C][C]^T \quad , \tag{15}$$

where [C] is a lower triangular matrix. This decomposition is performed by subroutine SDCØMP (see Section 3.5.14 of the Programmer's Manual).

The symmetrical [J] matrix is then obtained by the following transformation of the eigenvalue problem. Premultiply Equation 1 by $[C]^{-1}$ and substitute for $[M_{aa}]$ in Equation 15.

$$[C]^{-1} [K_{aa}]\{u_a\} - \lambda[C]^{-1}[C][C]^T \{u_a\} = 0 \quad . \tag{16}$$

Let

$$\{u_a\} = [C]^{-1,T} \{w\} \quad , \tag{17}$$

where {w} is called the transformed vector. Equation 16 then reduces to Equation 14 with

$$[J] = [C]^{-1}[K_{aa}][C]^{-1,T} \quad . \tag{18}$$

After the transformed eigenvectors of Equation 14, $\{\phi_w\}$, have been calculated by the tridiagonal method they are converted to physical form by

$$\{\phi_a\} = [C]^{-1,T}\{\phi_w\} \quad . \tag{19}$$

In order that the Cholesky decomposition, Equation 15, be possible, it is required that $[M_{aa}]$ be a positive definite matrix. For many problems $[M_{aa}]$ would naturally be singular, as for example, when rotary inertias at rotational coordinates are zero. In these cases the user should request that the Guyan reduction, Section 3.5.4, be used to eliminate the massless degree of freedom, if the tridiagonal method is used.

The inversion of the [C] matrix is not difficult to perform since [C] is triangular. In addition $[C]^{-1}$ will be banded if there is no mass coupling between grid points and if the Guyan reduction has not been used extensively.

Once the finite frequency eigenvectors, $\{\phi_{af}\}$, have been calculated by the selected method, they are merged with the zero frequency eigenvectors, $\{\phi_{ao}\}$, to form the complete modal matrix of eigenvectors

$$[\phi_{ai}] = [\phi_{ao} \mid \phi_{af}] \quad . \tag{20}$$

The last operations performed by the real eigenvalue analysis module are to normalize the eigenvectors and to perform the mass orthogonality test, if it has been requested by the user. The test requires that

$$m_{ij} = \{\phi_{ai}\}^T [M_{aa}]\{\phi_{aj}\} < \varepsilon \quad , \quad i \neq j \quad , \tag{21}$$

where ε is supplied by the user, and $\{\phi_{ai}\}$ has been normalized to unit generalized mass, i.e.,

$$\{\phi_{ai}\}^T [M_{aa}]\{\phi_{ai}\} = 1 \quad . \tag{22}$$

If the test fails, the program provides the number of mode pairs failing the test and the value of the largest off-diagonal term.

It is recognized that the eigenvectors of extremely close or identical eigenvalues may be substantially correct even though they do not pass the orthogonality test. Eigenvectors obtained with the inverse power method are orthogonalized with respect to all previously extracted eigenvalues. If the determinant method has been selected a preliminary test on the closeness of eigenvalues

$$\frac{\lambda_i - \lambda_j}{\lambda_i + \lambda_j} < .01 \quad , \tag{23}$$

is made. If a group of adjacent eigenvalues satisfy the closeness condition, the orthogonality test, Equation 21, is applied to the group. The eigenvectors of the group that fail the orthogonality test are then orthogonalized by the Schmidt procedure

$$\{\bar{\phi}_{ai}\} = \{\phi_{ai}\} - \sum_{j=1}^{i-1} m_{ij}\{\bar{\phi}_{aj}\} \quad , \tag{24}$$

where $\{\bar{\phi}_{ai}\}$ is a "purified" eigenvector, and the sum extends over preceding members of the group. Once the purification of close eigenvectors is completed, the complete mass orthogonality test is performed.

The user may request any one of the following forms of normalization for the eigenvectors

1. Normalize to unit generalized mass.
2. Normalize so that the largest element of the vector is unity.
3. Normalize so that a particular element of the vector is unity.

The generalized mass, $\{\phi_{ai}\}^T [M_{aa}]\{\phi_{ai}\}$, is included in the output for each eigenvalue.

It may be noted here that, in the case of non-multiple eigenvalues, each of the four extraction methods will, for a given type of normalization, give essentially the same eigenvectors. However, in the case of multiple eigenvalues, the four methods will, in general, give different eigenvectors even though they may employ the same type of normalization. This discrepancy may seem disturbing, but it is only apparent and not real. This is due to the fact that the arbitrary constants that are inherently assumed in the computation of eigenvectors are, in general, different for the different methods. It can be easily shown that the apparently different eigenvectors of multiple eigenvalues obtained by different extraction methods have certain definite relationships among them. This is fully explained in Reference 2.

9.2.2 The APPEND Feature for Real Eigenvalue Analysis

In real eigenvalue analysis, it is frequently necessary to add new eigenvalues and eigenvectors to those already computed in a previous run. A capability called the APPEND feature makes it possible to do this without re-executing the entire problem. It is available only when using the

Inverse Power, Determinant and Tridiagonal Reduction (FEER) methods of eigenvalue extraction in Rigid Formats 3, 10, 11, 12, 13, and 15 (Displacement Approach) and in Rigid Formats 10 and 11 (Aeroelastic Approach).

In order to use the APPEND feature, the user first requests a checkpoint of an eigenvalue problem. This run can terminate for any reason so long as the READ module finds at least one eigenvalue and eigenvector and the LAMA (eigenvalue) and PHIA (eigenvector) files are successfully checkpointed. The READ module also sets the parameter NEIGV to be equal to the number of eigenvalues and eigenvectors found in this checkpoint run.

The user then restarts and activates the APPEND feature by changing either the METHØD card in the Case Control Deck and/or the EIGR card in the Bulk Data Deck so as to force the re-execution of the READ module.* The method of eigenvalue extraction used in the restart need not be the same as that used in the checkpoint run, but the structural model and the constraint data must be the same. Also, the user must ensure that the range of eigenvalues specified on the EIGR Bulk Data card for the restart does not include the eigenvalues that have previously been found and checkpointed. It is left to the user to satisfy this requirement. The program does not check for this condition.

The APPEND feature causes the READ module to retrieve the specified number (the first NEIGV) of eigenvalues and eigenvectors from the previously checkpointed LAMA and PHIA files (this retrieval is done in subroutine READ7 within the READ module; the number of eigenvalues and eigenvectors retrieved is indicated by a user information message) and to subsequently combine them with the newly computed results. (See the flow diagram in Figure 1.) The eigenvalues and eigenvectors output by the restart include those retrieved from the previously checkpointed run. Also, the resulting eigenvectors are normalized according to the method of normalization specified in the restart.

9.2.3 Complex Eigenvalue Analysis

The form of the complex eigenvalue problem using a direct formulation is

$$[M_{dd}\, p^2 + B_{dd}\, p + K_{dd}]\{u_d\} = 0. \tag{25}$$

---

*Additionally, if the user wishes to retrieve only the first n (n < NEIGV) (rather than all the NEIGV) eigenvalues and eigenvectors found on a checkpoint run, the parameter NEIGV should be reset to n in the restart by means of a PARAM statement just before the READ module in the DMAP sequence. This is done by means of a DMAP alter in the Executive Control Deck of the restart.

The vector $\{u_d\}$ includes the set, u_a, of degrees of freedom at structural grid points and the set of extra points, u_e, that are described in Section 9.3. The elements of the mass matrix, $[M_{dd}]$, the damping matrix, $[B_{dd}]$, and the stiffness matrix, $[K_{dd}]$, may be real or complex and the matrices may be symmetric or nonsymmetric, singular, or nonsingular. The eigenvalue, p_j, corresponds to a homogeneous solution of Equation 25,

$$\{u_d\} = \{\phi_{dj}\} e^{p_j t} \quad , \tag{26}$$

or

$$\{u_d\} = \{\phi_{dj}\} e^{\alpha_j t} \sin(\omega_j t) \quad , \tag{27}$$

where α_j is the real part of p_j and ω_j is the imaginary part.

The form of the complex eigenvalue problem using a modal formulation is

$$[M_{hh} p^2 + B_{hh} p + K_{hh}]\{u_h\} = 0 \quad . \tag{28}$$

The components of $\{u_h\}$ are the set of modal coordinates, ξ_i, and the set of extra points, u_e, (see Section 9.3). As in the case of the direct formulation, there are no restrictions on the matrices in Equation 28.

Two optional methods of eigenvalue extraction, the inverse power method with shifts, and the determinant method, are provided. They are described in Section 10.

The eigenvectors are normalized to a maximum element value of unity, or to a value of unity for a specified element, according to user's option. Generalized modal masses are not calculated and orthogonality tests, if such could be defined, are not performed.

REFERENCES

1. Hildebrand, F. B., Methods of Applied Mathematics, Prentice-Hall, New York, 1952, pg. 35.

2. Pamidi, P. R., and Brown, W. K., "On Eigenvectors of Multiple Eigenvalues Obtained in NASTRAN," NASTRAN: Users' Experiences, NASA Technical Memorandum, NASA TM X-3278, September 1975, pp. 285-300.

ORIGINAL PAGE IS OF POOR QUALITY

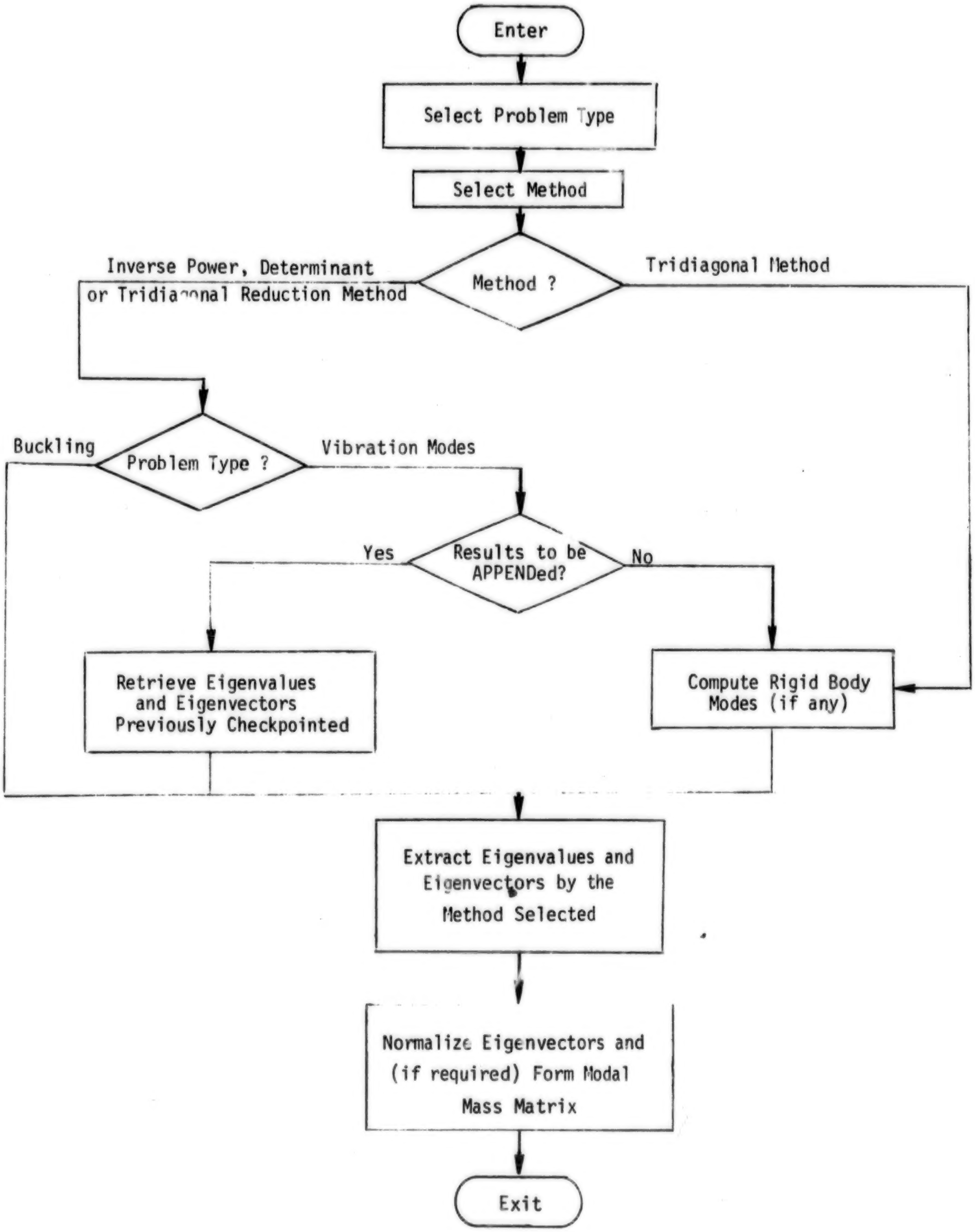

Enter
Select Problem Type
Select Method
Method ?
Inverse Power, Determinant or Tridiagonal Reduction Method
Tridiagonal Method
Buckling
Problem Type ?
Vibration Modes
Yes
Results to be APPENDed?
No
Retrieve Eigenvalues and Eigenvectors Previously Checkpointed
Compute Rigid Body Modes (if any)
Extract Eigenvalues and Eigenvectors by the Method Selected
Normalize Eigenvectors and (if required) Form Modal Mass Matrix
Exit

Figure 1. Flow diagram for the real eigenvalue analysis module, READ.

9.3 ASSEMBLY OF DYNAMIC MATRICES

The matrix properties of the dynamic system are assembled in three different modules, as shown in Figure 1 of Section 9.1. The modules are

1. DPD, The Dynamic Pool Distributor
2. GKAD, The Direct Dynamic Matrix Assembler
3. GKAM, The Modal Dynamic Matrix Assembler

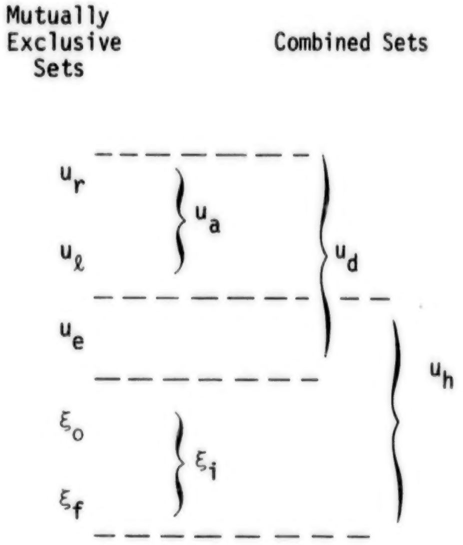
The Dynamic Pool Distributor performs a number of preliminary bookkeeping chores. It generates flags defining members of various displacement sets; it generates tables relating internal and external grid point numbers, including Extra Points introduced explicitly for dynamic analysis; it organizes Transfer Function data and Eigenvalue Extraction data; it prepares tables for Dynamic Loads and Nonlinear Functions; and it compiles lists of response quantities for Transient Analysis.

The function of the Direct Dynamic Matrix Assembler is to assemble the mass, damping, and stiffness matrices from various sources in terms of displacements at grid points. The function of the Modal Dynamic Matrix Assembler is to apply a modal transformation to the mass, damping, and stiffness matrices.

Other modules performing functions in connection with dynamic system assembly are the Frequency Response Analysis Module, which assembles loads, and the Transient Response Analysis Module, which assembles loads and nonlinear terms in the equations of motion. Discussion of these functions is included with the discussion of the other functions of the modules(see Sections 11 and 12).

9.3.1 Notation System

Before proceeding to a detailed description of the process of matrix assembly, it is well to review the notation system used in dynamic analysis, which was briefly described in Section 3.3. The main differences from static analysis are the addition of modal coordinates, ξ_i, and the addition of extra points, u_e. The nesting of the displacements sets used in dynamic analysis is depicted by the following diagram.



The definitions of the mutually exclusive sets are

u_r	points to which determinate reactions are applied in static analysis,
u_ℓ	the remaining structural points (points left over) which are evaluated in static analysis by direct solution of the stiffness matrix,
u_e	extra degrees of freedom introduced in dynamic analysis to describe control systems, etc.,
ξ_o	rigid body (zero frequency) modal coordinates,
ξ_f	finite frequency modal coordinates.

The combined displacement sets are

$u_a = u_r + u_\ell$, the set used in real eigenvalue analysis,

$\xi_i = \xi_o + \xi_f$, the set of all modal coordinates,

$u_d = u_a + u_e$, the set used in dynamic analysis by the direct method,

$u_h = \xi_i + u_e$, the set used in dynamic analysis by the modal method.

Load vectors and two-dimensional arrays employ the subscripts of the displacement sets with which they are associated. Occasionally additional subscripts are used to distinguish between two members of the same set. Superscripts are used to distinguish different kinds of entities. See Section 3.3 for further details.

9.3.2 Extra Points and Transfer Functions

In NASTRAN dynamic analysis is performed via a basic quadratic format,

$$[Mp^2 + Bp + K]\{u\} = \{P\} \quad . \tag{1}$$

Situations occur, rather frequently, where some of the properties of a dynamic system cannot be expressed directly as constant coefficients in a mass, damping, or stiffness matrix. If the problem is formulated in the frequency domain, as are many problems in aeroelasticity and hydroelasticity, terms may occur that have other than constant, linear or quadratic dependence on p. These terms may, however, be expressed as frequency-dependent coefficients in the mass, damping, or stiffness matrices. This creates a certain awkwardness in eigenvalue extraction, but it is not a serious inconvenience in frequency response analysis.

If the problem is formulated in the time domain, in which case p is a derivative operator, the nonconforming properties cannot, even conceptually, be treated as time-dependent coefficients. There are two types of nonconformities: those in which the resulting forces are nonlinear functions of displacements and of their derivatives; and those in which the resulting forces are linear functions of higher order derivatives, or, in general, are convolution integrals of the displacements. The former type of nonconformity, which is included only in transient analysis, is discussed in Section 11.2. The latter type is discussed here.

A convenient representation of a convolution integral is its transfer function in the frequency domain, i.e., its Laplace transform. The properties of control system components, or even of complete control systems, are most frequently described by transfer functions, represented by algebraic expressions of the form

$$e_2 = H_{12}(p)e_1 \quad , \tag{2}$$

where

$$H_{12}(p) = \frac{a_o + a_1p + a_2p^2 + \ldots + a_np^n}{b_o + b_1p + b_2p^2 + \ldots + b_mp^m} \tag{3}$$

e_1 and e_2 may be variables representing voltages in amplifier circuits, valve positions in hydraulic actuators, etc. All such quantities are treated as degrees of freedom and are included in the displacement vector $\{u_e\}$ (extra points). Transfer functions, Equation 3, are brought within the basic quadratic format as follows.

If $n \leq 2$ and $m \leq 2$, then Equation 2 may be written

$$(b_2p^2 + b_1p + b_o)e_2 - (a_2p^2 + a_1p + a_o)e_1 = 0 \quad , \tag{4}$$

which may properly be regarded as a differential equation of second order, $(p = d/dt)$. The coefficients b_2, b_1, and b_o are, respectively, the diagonal terms of the [M], [B], and [K] matrices in the row reserved for e_2. The terms $-a_2$, $-a_1$, and $-a_o$ are, respectively, those in the row corresponding to e_2 and the column corresponding to e_1 in the [M], [B], and [K] matrices.

If $2 < n \leq 4$ or $2 < m \leq 4$, the transfer function $H_{12}(p)$ may be factored into a pair of quadratic fractions such that

$$\begin{aligned} H_{12}(p) &= \left(\frac{a_o{}^a + a_1{}^a p + a_2{}^a p^2}{b_o{}^a + b_1{}^a p + b_2{}^a p^2}\right) \cdot \left(\frac{a_o{}^b + a_1{}^b p + a_2{}^b p^2}{b_o{}^b + b_1{}^b p + b_2{}^b p^2}\right) \\ &= H^{(a)}(p) \cdot H^{(b)}(p) \quad . \end{aligned} \tag{5}$$

The transfer function is then brought within the required format by defining another extra point, e_3, such that

$$\left.\begin{aligned} e_3 &= H^{(a)}(p) \cdot e_1 \quad , \\ e_2 &= H^{(b)}(p) \cdot e_3 \quad . \end{aligned}\right\} \tag{6}$$

The extension to higher order polynomials is evident. The factoring of polynomials, which will not often be required, is left to the user. An alternative reduction of the high order transfer function is provided by fraction series expansion:

$$H_{12} = \sum_i \left(\frac{a_0^{\,i} + a_1^{\,i}p + a_2^{\,i}p^2}{b_0^{\,i} + b_1^{\,i}p + b_2^{\,i}p^2}\right) = \sum_i H^{(i)}(p) \quad . \tag{7}$$

which has the advantage of not requiring the numerator to be factored.

In this case auxiliary variables e_i are introduced such that

$$e_i = H^{(i)}(p)e_1 \quad , \quad i = 3,\ 4,\ 5,\ \text{etc.}, \tag{8}$$

and

$$e_2 = \sum_i e_i \quad . \tag{9}$$

Equation 8 is treated as a set of differential equations, as before. Equation 9 is also treated as a differential equation where the only non-zero matrix elements are in the row of K corresponding to e_2.

In order to facilitate the treatment of control systems, NASTRAN includes an input data format for the specification of transfer functions in the form

$$u_e = \frac{1}{b_0 + b_1p + b_2p^2} \sum_i \left(a_0^{\,i} + a_1^{\,i}p + a_2^{\,i}p^2\right) u_i \quad , \tag{10}$$

which is interpreted by the program as a differential equation with terms in the rows of [M], [B], and [K] corresponding to u_e. u_e must be an extra point, but u_i can either be an extra point or any structural point contained in u_g.

Structural loads proportional to displacements at extra points are represented by adding terms in the stiffness matrix at the intersections of rows corresponding to the structural points and columns corresponding to the extra points. Direct input data cards are provided for this purpose. The superposition of all such terms, including those generated by transfer functions, is called the direct input stiffness matrix, $[K^2]$. The direct input mass and damping matrices are $[M^2]$ and $[B^2]$. Terms in the direct input matrices may refer to extra points, to structural points or to both. Since dynamic loads may also be applied to extra points, the means are available for the complete simulation of subsystems by means of extra points. Some applications to control system simulation are described in Section 14.2.

9.3.3 Direct Dynamic Matrix Assembly

The Direct Dynamic Matrix Assembler, GKAD, performs three main functions.

1. It decodes transfer function information and adds it to the direct input matrices $[M_{pp}{}^2]$, $[B_{pp}{}^2]$, and $[K_{pp}{}^2]$ as explained above.
2. It reduces the direct input matrices by the application of constraints and partitioning from the displacement set, u_p, (all physical points) to the u_d set. See Section 3.3.
3. It assembles the complete dynamic matrices $[M_{dd}]$, $[B_{dd}]$, and $[K_{dd}]$ from the direct input matrices and from the structural matrices generated in the statics portion of the program.

Direct input matrices are, at present, generated by transfer functions or they are supplied directly by the user. It is possible to modify the program so as to provide internal calculation of direct input matrices corresponding to particular aeroelastic and hydrodynamic environments.

The extra points are carried along in the application of constraints and partitioning to the direct input matrices, at the positions in u_p indicated in the grid point sequence list. For this purpose the multipoint constraint matrix $[G_m]$ (see Section 3.5.1), and the Guyan reduction matrix $[G_o]$, (see Section 3.5.4), are expanded to include the u_e coordinates. The corresponding columns are, of course, null.

The specific steps in the reduction as applied to $[K_{pp}{}^2]$ are:

1. Eliminate multipoint constraints.

$$[K_{nn}{}^2] = [\bar{K}_{nn}{}^2] + [K_{nm}{}^2][G_m] + [G_m]^T [K_{mn}{}^2] + [G_m]^T [K_{mm}{}^2][G_m] \quad , \qquad (11)$$

where the partitions of $[K_{pp}{}^2]$ are given by

$$[K_{pp}{}^2] = \left[\begin{array}{c|c} \bar{K}_{nn}{}^2 & K_{nm}{}^2 \\ \hline K_{mn}{}^2 & K_{mm}{}^2 \end{array}\right] \qquad (12)$$

(The u_e coordinates are included in the u_n coordinates.)

2. Eliminate single-point constraints by extracting $[K_{ff}^2]$ from

$$[K_{nn}^2] = \left[\begin{array}{c|c} K_{ff}^2 & K_{fs}^2 \\ \hline K_{sf}^2 & K_{ss}^2 \end{array}\right] . \qquad (13)$$

(The u_e coordinates are included in the u_f coordinates.)

3. Eliminate the omitted coordinates, u_o, via the Guyan reduction

$$[K_{dd}^2] = [\overline{K}_{dd}^2] + [K_{do}^2][G_o] + [G_o]^T[K_{od}^2] + [G_o]^T[K_{oo}^2][G_o] \quad , \qquad (14)$$

where the partitions of $[K_{ff}^2]$ are given by

$$[K_{ff}^2] = \left[\begin{array}{c|c} \overline{K}_{dd}^2 & K_{do}^2 \\ \hline K_{od}^2 & K_{oo}^2 \end{array}\right] . \qquad (15)$$

(The u_d set is the union of u_a and u_e.)

The direct input mass and damping matrices are treated in an identical manner. When the elements of the direct input matrices are complex numbers, which they may be for frequency response and complex eigenvalue analysis, the real and imaginary parts are reduced at the same time.

The final assembly of the dynamic matrices is performed as follows. The structural matrices $[K_{aa}]$, $[K_{aa}^4]$, $[M_{aa}]$, and $[B_{aa}]$ are expanded by the addition of zeroes in the rows and columns corresponding to extra points, to form $[K_{dd}^1]$, $[K_{dd}^4]$, $[M_{dd}^1]$, and $[B_{dd}^1]$. $[K_{aa}^4]$ is a structural damping matrix obtained by multiplying the stiffness matrix $[K_e]$ of an individual structural element by a damping factor, g_e, and combining the results for all such structural elements. Each element may have a different damping factor. $[B_{aa}]$ is a viscous damping matrix resulting from viscous rod elements (extension or twist) and viscous scalar dampers.

For frequency response analysis and complex eigenvalue analysis the complete dynamic matrices are

$$[K_{dd}] = (1 + ig)[K_{dd}^1] + [K_{dd}^2] + i[K_{dd}^4] \quad , \qquad (16)$$

$$[B_{dd}] = [B_{dd}{}^1] + [B_{dd}{}^2] \quad , \tag{17}$$

$$[M_{dd}] = [M_{dd}{}^1] + [M_{dd}{}^2] \quad . \tag{18}$$

$[K_{dd}{}^1]$ is multiplied by the factor $(1 + ig)$ to include uniform structural damping in cases where it is appropriate. The constant g is specified by the user.

For transient response analysis the complete dynamic matrices are

$$[K_{dd}] = [K_{dd}{}^1] + [K_{dd}{}^2] \quad , \tag{19}$$

$$[B_{dd}] = [B_{dd}{}^1] + [B_{dd}{}^2] + \frac{g}{\omega_3}[K_{dd}{}^1] + \frac{1}{\omega_4}[K_{dd}{}^4] \quad , \tag{20}$$

$$[M_{dd}] = [M_{dd}{}^1] + [M_{dd}{}^2] \quad . \tag{21}$$

The constant ω_3 is the radian frequency at which the term $\frac{g}{\omega_3}[K_{dd}{}^1]$ produces the same damping forces as the term $ig[K_{dd}{}^1]$ in frequency response analysis. The viscous damping forces are larger at higher frequencies and smaller at lower frequencies. A small value of g/ω_3 is frequently useful to ensure stability of higher modes in nonlinear transient analysis. The user selects the values of ω_3 and ω_4. If they are omitted the corresponding terms are ignored.

9.3.4 Modal Dynamic Matrix Assembly

The Modal Dynamic Matrix Assembler, GKAM, assembles the dynamic matrices $[M_{hh}]$, $[B_{hh}]$, and $[K_{hh}]$ in the general dynamic equation

$$[M_{hh}\, p^2 + B_{hh}\, p + K_{hh}]\{u_h\} = \{P_h\} \quad . \tag{22}$$

The modal analysis coordinates, u_h, are related to the corresponding physical coordinates, u_d, as follows. The set u_h is the union of the modal coordinates ξ_i and the extra points u_e

$$\left\{\begin{array}{c}\xi_i \\ \hline u_e\end{array}\right\} \longrightarrow \{u_h\} \quad . \tag{23}$$

623

while u_d is the union of u_a and u_e. The transformation between ξ_i and u_a is

$$\{u_a\} = [\phi_{ai}]\{\xi_i\} \quad , \tag{24}$$

where $[\phi_{ai}]$ is the matrix of eigenvectors obtained in real eigenvalue analysis, (see Section 9.2.1).

The transformation from u_h to u_d is obtained by augmenting $[\phi_{ai}]$ to include the extra points. Thus

$$\{u_d\} = [\phi_{dh}]\{u_h\} \quad , \tag{25}$$

where

$$[\phi_{dh}] = \left[\begin{array}{c|c} \phi_{ai} & 0 \\ \hline 0 & I \end{array}\right] \quad . \tag{26}$$

$[\phi_{dh}]$ is actually in this form because the components of u_e are placed last in u_h and also in u_d for modal analysis.

The dynamic system properties, exclusive of loads, included in dynamic analysis by the modal method are the direct input matrices $[M_{dd}{}^2]$, $[B_{dd}{}^2]$, and $[K_{dd}{}^2]$, obtained from the Direct Dynamic Matrix Assembler, and the modal matrices, $[m_i]$, $[b_i]$, and $[k_i]$, obtained from the Real Eigenvalue Analysis module. $[m_i]$ is the modal mass matrix defined by Equation 2 of Section 9.2.1, with off-diagonal terms (which should be zero anyway) omitted. The damping matrix $[b_i]$ and the stiffness matrix $[k_i]$ are obtained from $[m_i]$ by

$$[b_i] = [\omega_i \; g(\omega_i) m_i] \quad , \tag{27}$$

$$[k_i] = [\omega_i{}^2 \; m_i] \quad , \tag{28}$$

where ω_i is the radian frequency of the ith normal mode and $g(\omega_i)$ is a damping factor. $g(\omega_i)$ is obtained by interpolation of a table, $g(\omega)$, supplied by the user to represent the variation of structural damping with frequency.

It will be noted that the specification of damping properties for the modal method is somewhat less general than it is for the direct method in that viscous dampers $[B_{aa}]$ and nonuniform structural damping $[K_{aa}{}^4]$ are not used. The damping function $g(\omega)$ is regarded as an adequate substitute.

The complete dynamic matrices for the modal method of dynamic analysis are obtained from

$$[M_{hh}] = \left[\begin{array}{c|c} m_i & 0 \\ \hline 0 & 0 \end{array}\right] + [\phi_{dh}]^T\,[M_{dd}{}^2][\phi_{dh}] \quad , \qquad (29)$$

$$[B_{hh}] = \left[\begin{array}{c|c} b_i & 0 \\ \hline 0 & 0 \end{array}\right] + [\phi_{dh}]^T[B_{dd}{}^2][\phi_{dh}] \quad , \qquad (30)$$

$$[K_{hh}] = \left[\begin{array}{c|c} k_i & 0 \\ \hline 0 & 0 \end{array}\right] + [\phi_{dh}]^T[K_{dd}{}^2][\phi_{dh}] \quad . \qquad (31)$$

The user can specify the number of modes, starting from zero frequency and including rigid body modes, or he can specify that all the modes within a prescribed frequency range be used.

A useful feature of NASTRAN is that the user can specify the structural properties of parts of the structure by means of scalar elements and constraints whose values are derived from substructure modes (see Section 14.1). The program treats such elements in the same manner as all other elements, so that their effects are included in $[K_{aa}]$, $[M_{aa}]$, and $[B_{aa}]$.

9.4 DYNAMIC DATA RECOVERY

The dynamic analysis modules for complex eigenvalue analysis, frequency response analysis, and transient response analysis produce either $\{u_d\}$ or $\{u_h\}$ as solution vectors, depending on whether a direct or a modal formulation has been employed. The transient response module also produces the first and second time derivatives of the solution vectors. The additional operations in the data recovery chain are indicated in Figure 1 of Section 9.1. The solution vector is partitioned, transformed, and augmented in various ways to obtain desired output quantities.

The main function of the Dynamic Data Recovery module (DDR), see Figure 1 of this section, is to produce a displacement vector, $\{u_a\}$, which can be processed by the static data recovery modules to produce the complete structural displacement vector, $\{u_g\}$, and the internal forces and stresses in selected structural elements (see Section 3.7). In the standard method of dynamic data recovery, which was the only method available prior to Level 16 of NASTRAN, the static data recovery modules process the solution vector for each mode, time step, or frequency as though it were a static solution vector, i.e., all calculations are repeated for every vector. Beginning with Level 16 of NASTRAN, a second method, called the Matrix Method of Dynamic Data Recovery, has been added for transient response and frequency response which is considerably more efficient when the number of time steps or frequencies exceeds the number of independent degrees of freedom. In this method a matrix is precomputed which relates the desired output quantities of whatever type directly to the independent degrees of freedom. Thus, in direct analysis,

$$\{u_j\} = [S_{jd}]\{u_d\} \quad , \tag{1}$$

and in modal analysis,

$$\{u_j\} = [S_{jh}]\{u_h\} \quad . \tag{2}$$

$[S_{jd}]$ and $[S_{jh}]$ are matrices of constant coefficients and $\{u_j\}$ is the vector of desired output quantities, which may include displacements, velocities, accelerations, internal forces, stresses, and forces of constraint. Note that the relationship of output quantities to independent degrees of freedom is linear so that nonlinear stress resultants, such as principal stresses, are excluded.

The matrix $[S_{jd}]$ is formed in two steps. First, the columns of an identity matrix of order $(d \times d)$ are processed sequentially by module SDR1, which recovers the degrees of freedom eliminated by constraints (see Section 3.7.1), to form a matrix $[u_{pd}]$ in which each column is the vector of grid point displacements, extra point displacements, and single point forces of constraint due to a unit value of one dynamic degree of freedom. Second, the columns of $[u_{pd}]$ are processed sequentially by module SDR2 (see Section 3.7.2) to form a matrix $[\sigma_{ed}]$ in which each column is the vector of user-requested stresses and internal forces in elements due to unit value of one dynamic degree of freedom. The union of $[\sigma_{ed}]$ with user-requested rows of $[u_{pd}]$ form the matrix $[S_{jd}]$.

The matrix $[S_{jh}]$ is formed in a similar way except that in the first step the identity matrix is replaced by $[\phi_{dh}]$, the augmented modal matrix (see Section 9.3.4).

If the number of dynamic degrees of freedom is less than the number of time steps or frequencies, the matrix method of dynamic data recovery will be more efficient than the standard method. The saving may be very great, particularly in the case of the modal formulation when the number of modes is small.

In the event that it is desired to print or plot each output quantity versus time or frequency, it is required that the matrix of output quantities, $[u_j]$, be transposed. This is actually performed by transposing the output data recovery matrix first, followed by the calculation

$$[u_j]^T = [u_d]^T[S_{jd}]^T \quad \text{or} \quad [u_j]^T = [u_h]^T[S_{jh}]^T \quad . \tag{3}$$

One of the two methods for multiplying $[A]^T[B]$ described on page 2.1-4 is used. Both methods avoid the transposition of $[u_h]$ or $[u_d]$.

In the mode acceleration method, which is only available as a subcase of the standard method (see Figure 1), the vector $\{u_a\}$ that describes unconstrained motions of the structure is replaced by another vector $\{u_a^a\}$ that will produce more accurate estimates of the stresses. The motivation stems from the fact that when modal analysis is employed, the modal vector $\{\xi_i\}$ that replaces $\{u_a\}$ is nearly always a smaller set from which higher modes have been omitted. The influence of the higher modes can be approximated by observing that their response to low frequency dynamic excitation is almost purely static. Thus, the inertia and damping forces on the structure contain very little contribution from the higher modes. In the mode acceleration method, the inertia and damping forces are computed from the modal solution. These forces are then added to the applied forces and are used to obtain a more accurate displacement vector for the structure by static analysis. The details of the computation are as follows.

When the general dynamic system is formed using modes, the vector $\{u_h\}$ used in the analysis contains the modal coordinates $\{\xi_i\}$ and the extra degrees of freedom $\{u_e\}$. The system from which $\{\xi_i\}$ is abstracted consists of the stiffness matrix $[K_{aa}]$, the mass matrix $[M_{aa}]$, and modal damping elements, b_i. The presence of the modal damping elements is ignored in data recovery. The general system includes, in addition, direct input matrices $[M^2]$, $[B^2]$, and $[K^2]$ and loads $\{P_a\}$ and $\{P_e\}$ applied, respectively, to points $\{u_a\}$ and $\{u_e\}$.

In order to apply the mode acceleration method, the system that existed prior to the application of modal constraints is reconstructed, as follows:

$$[M_{dd}\,p^2 + B_{dd}{}^2\,p + (K_{dd}{}^1 + K_{dd}{}^2)]\{\bar{u}_d\} = \{P_d\} \quad . \tag{4}$$

$\{\bar{u}_d\}$ differs from $\{u_d\}$, the solution vector actually obtained, due to the presence of higher modes. Note the presence of $[B_{dd}{}^2]$ from Equation 30 of Section 9.3.4.

The elastic properties of the structure are represented by $[K_{dd}{}^1]$ which is the structural stiffness matrix $[K_{aa}]$ augmented by zeroes in the rows and columns corresponding to the extra points, u_e, (see Section 9.3.3). $[K_{dd}{}^1]$ represents the complete structure as opposed to the modal stiffness matrix $[k_i]$ which ignores the higher deformation modes. The viewpoint is taken that the transformed solution vector, $\{u_d\} = [\phi_{dh}]\{u_h\}$, is adequate to evaluate the inertia (and damping) loads on the structure but that stresses derived from it could be improved.

The improved solution is the vector $\{u_d^a\}$ obtained by solving

$$[K_{dd}{}^1]\{u_d^a\} = \{P_d^a\} \quad , \tag{5}$$

where, comparing Equations 4 and 5

$$\{P_d^a\} = \{P_d\} - [M_{dd}\,p^2 + B_{dd}{}^2\,p + K_{dd}{}^2]\{u_d\} \quad . \tag{6}$$

The derivative operator p in Equation 6 is simply a number for complex eigenvalue and frequency response analysis. For transient analyses the velocity and acceleration vectors, $\{pu_d\}$ and $\{p^2\,u_d\}$, are contained in the output of the transient analysis module.

The procedure indicated in Equations 5 and 6 is modified in the program by the removal of extra points, u_e, and by the use of a special procedure for free bodies. First $\{P_d^a\}$ and $\{u_d^a\}$ are partitioned

$$\{P_d^a\} = \left\{\begin{array}{c} P_a^a \\ \hline P_e \end{array}\right\} \tag{7}$$

$$\{u_d^a\} = \left\{\begin{array}{c} u_a^a \\ \hline u_e \end{array}\right\} \quad . \tag{8}$$

Then, if there are no free body motions, $\{u_a^a\}$ is obtained by solving

$$[K_{aa}]\{u_a^a\} = \{P_a^a\} \quad . \tag{9}$$

by the standard procedures used in static analysis. Note that for complex eigenvalue extraction and for frequency response analysis $\{P_a^a\}$ may be complex, in which event the real and imaginary parts are considered as separate loading conditions. The operations of the static data recovery module are then applied separately to the real and imaginary parts of $\{u_a^a\}$.

If free body modes are present, $[K_{aa}]$ is singular, indicating the need for an alternate procedure. It is assumed that the motions at a set of fictitious reaction points, u_r, (see Section 3.5.5) are correctly given by the modal solution. The elastic deformations of the remaining points, u_ℓ, relative to u_r are obtained by solving

$$[K_{\ell\ell}]\{u_\ell^a\} = \{P_\ell^a\} \quad , \tag{10}$$

where $\{P_\ell^a\}$ is the appropriate partition of

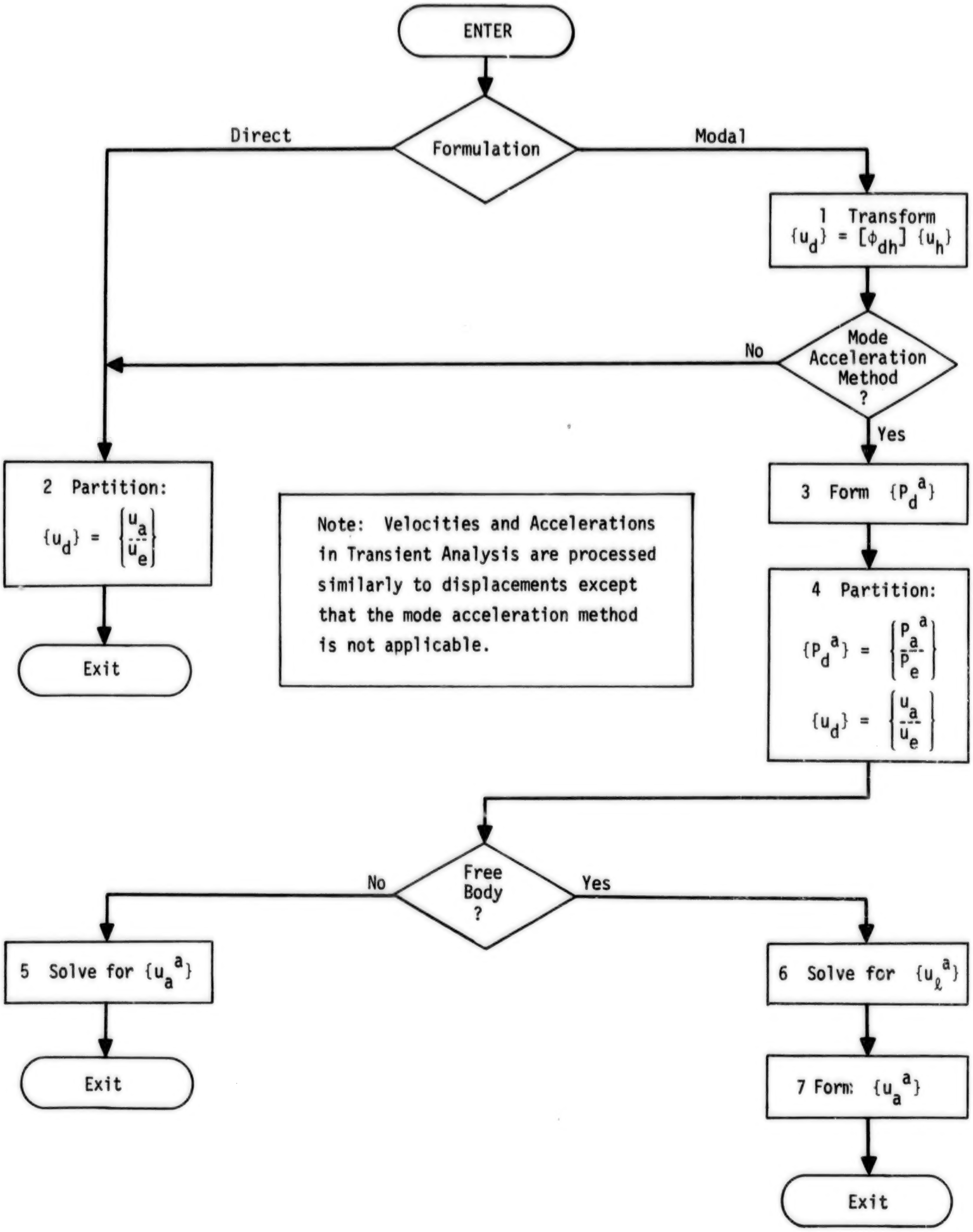
$$\{P_a^a\} = \left\{\begin{array}{c} P_\ell^a \\ \hline P_r^a \end{array}\right\} \quad . \tag{11}$$

The improved solution vector is obtained from

$$\{u_a^a\} = \left\{\begin{array}{c} u_\ell^a + D\,u_r \\ \hline u_r \end{array}\right\} \quad , \tag{12}$$

where [D] is the matrix that evaluates the rigid body motions of the structure (see Section 3.5.5).

Although the mode acceleration technique will produce better stresses, it may be expensive, particularly in transient analyses involving many time steps. Its use must, therefore, be separately justified for each application.


ENTER
Formulation
Direct
Modal
1 Transform $\{u_d\} = [\phi_{dh}] \{u_h\}$
Mode Acceleration Method ?
No
Yes
2 Partition: $\{u_d\} = \begin{Bmatrix} u_a \\ \hdashline u_e \end{Bmatrix}$
Exit
3 Form $\{P_d{}^a\}$
4 Partition: $\{P_d{}^a\} = \begin{Bmatrix} P_a{}^a \\ \hdashline P_e \end{Bmatrix}$ $\{u_d\} = \begin{Bmatrix} u_a \\ \hdashline u_e \end{Bmatrix}$
Free Body ?
No
Yes
5 Solve for $\{u_a{}^a\}$
Exit
6 Solve for $\{u_\ell{}^a\}$
7 Form: $\{u_a{}^a\}$
Exit

Note: Velocities and Accelerations in Transient Analysis are processed similarly to displacements except that the mode acceleration method is not applicable.

Figure 1. Dynamic Data Recovery, Standard Method

10. EIGENVALUE EXTRACTION METHODS

10.1 INTRODUCTION

Four methods of eigenvalue extraction are provided with NASTRAN because no single method, or pair of methods, has been found that is satisfactory with respect to efficiency, reliability, and generality of application in all situations. We are living in a time of rather explosive growth, engendered by the digital computer, in the variety and capability of eigenvalue extraction routines. New methods will be added to NASTRAN and old methods will be improved as discoveries are made.

Most methods of algebraic eigenvalue extraction belong to one of two groups - transformation methods and "tracking" methods. In a transformation method the matrix of coefficients is first transformed, while preserving its eigenvalues, into a special form (diagonal, tridiagonal, or upper Hessenberg) from which eigenvalues may easily be extracted. In a "tracking" method the roots are extracted, one at a time, by iterative procedures applied to the original dynamic matrix . Two of the methods used in NASTRAN are transformation methods (Tridiagonal Method and Tridiagonal Reduction or FEER Method), and the other two (Determinant Method and Inverse Power Method with Shifts) are tracking methods.

In the Tridiagonal Method, the full matrix of coefficients is converted into a condensed form, requiring that the major share of the total effort be expended prior to the extraction of the first eigenvalue. Thus, the total effort is not strongly dependent on the number of eigenvalues that are extracted. However, in the Tridiagonal Reduction Method, the matrix is not only converted to a condensed (tridiagonal) form, but it is simultaneously reduced in size. The size of the reduced matrix is of the same order of magnitude as the number of modes requested by the user. Therefore, this method possesses a major efficiency as compared to the straightforward Tridiagonal Method, namely that the effort expended is proportional to the number of extracted eigenvalues. The computations required in the tracking methods are also linearly proportional to the number of extracted eigenvalues and as a result they tend to be more efficient when only a few eigenvalues are required.

The general characteristics of the methods used in NASTRAN are compared in Table 1. The Tridiagonal Method, due to restrictions on matrix form, is available only for the evaluation of the vibration modes of conservative systems (see Section 9.2) and not for buckling or complex eigenvalue analysis. The Tridiagonal Reduction (FEER) Method, however, accommodates both vibration modes and buckling analyses. The other two methods are available for all real and complex eigenvalue problems currently solved by NASTRAN. The Determinant Method would, in addition, be applicable to future problem types in which the coefficients of the dynamic matrices are general functions of the eigenvalue.

It may be noted, from Table 1, that a narrow bandwidth, as well as a small proportion of extracted roots, tends to favor the tracking methods and the Tridiagonal Reduction Method. An example of such a problem is the evaluation of the lowest few modes of a launch vehicle.

When several modes of a real eigenvalue problem are required, the Tridiagonal Reduction Method will probably be the most efficient one to use, since it is a finite, non-iterative process usually requiring only one initial eigenvalue shift. When almost all the modes are required, the Tridiagonal Method may exceed the efficiency of the Tridiagonal Reduction Method, particularly if the bandwidth is relatively large. It should be noted, however, that when the order of the coefficient matrix exceeds two or three hundred, the Tridiagonal Method will probably require prohibitively time-consuming out-of-core operations, even on computers with relatively large central memory capacities, thereby losing its efficiency. This disadvantage does not apply to the Tridiagonal Reduction Method which is more efficient for out-of-core operations.

The Determinant Method and the Inverse Power Method with Shifts have the same general characteristics with respect to all current NASTRAN problems. The Inverse Power Method is, however, more efficient except when the bandwidth is extremely narrow as, for example, in beam problems. The main advantage of including both methods is the redundancy that is provided in case one method should fail (as sometimes happens with any method of eigenvalue extraction).

ORIGINAL PAGE IS
OF POOR QUALITY

Table 1. Comparison of Methods of Eigenvalue Extraction

Method Characteristics	Tridiagonal Method	Inverse Power Method With Shifts	Determinant Method	Tridiagonal Reduction Method
Most general form of matrix	$[A - pI]$	$[Mp^2 + Bp + K]$	$[A(p)]$	$[Mp^2 + K]$
Restrictions on matrix character	A real, sym., constant	M, B and K constant	None	M, K, real, sym., constant
Obtains eigenvalues in order	All at once	Nearest to shift point	(Usually) nearest to starting points	Vibration: Nearest to shift point Buckling: Lowest eigenvalue first
Takes advantage of bandwidth	No	Yes	Yes	Yes
Number of calculations, order of	$0(n^3)$	$0(nb^2E)$	$0(nb^2E)$	$0(n[b+E]^2)$

Note: n = number of equations

b = simi-bandwidth

E = number of eigenvalues extracted

624

10.2 THE TRIDIAGONAL METHOD

10.2.1 Introduction

The tridiagonal method is particularly effective for obtaining vibration modes when all, or a substantial fraction, of the eigenvalues and eigenvectors of a real symmetric matrix are desired. The general restrictions on the use of the method within NASTRAN are described in Section 10.1. The basic steps employed in the method are as follows. First, the eigenvalue problem,

$$[K_{aa} - \lambda M_{aa}]\{u_a\} = 0, \tag{1}$$

is transformed to the form

$$[A]\{x\} = \lambda\{x\}. \tag{2}$$

Second, [A] is converted, while preserving its eigenvalues, to a tridiagonal matrix by the Givens Method, Reference 1. Third, all the eigenvalues of the tridiagonal matrix are extracted using a modified Q-R algorithm. Fourth, the eigenvectors are computed over a given frequency range or for a given number of eigenvalues, and are converted to physical form. The transformation of the eigenvalue problem from Equation 1 to Equation 2 and the conversion of the eigenvectors to physical form are described in Section 9.2. A simplified flow diagram for the tridiagonal method is shown in Figure 1.

10.2.2 Tridiagonalization by the Givens Method

The most stable transformation methods of solution of eigenproblems for large symmetric matrices are based on the tridiagonalization techniques of Givens (Reference 1), Lanczos (Reference 2), and Householder (Reference 3). The Givens method, which is used by NASTRAN, depends on orthogonal transformations $[T]\,[A]\,[T]^T$ of a symmetric matrix [A]. An orthogonal transformation is one whose matrix [T] satisfies

$$[T][T]^T = [T]^T[T] = [I]. \tag{3}$$

The eigenvalues of a matrix are preserved under orthogonal transformation since

$$[T]\Big([A] - \lambda[I]\Big)[T]^T = [T][A][T]^T - \lambda[I]. \tag{4}$$

Consequently if det $\Big([A] - \lambda[I]\Big)$ vanishes, then det $\Big([T][A][T]^T - \lambda[I]\Big)$ also vanishes.

PAGE 626 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

626

The effect of a series of orthogonal transformations on the eigenvectors of a matrix is that of a succession of multiplications by orthogonal matrices. For if

$$[A]\{x\} = \lambda\{x\}, \tag{5}$$

and if $[T_1]$, $[T_2]$, ..., $[T_r]$ are orthogonal matrices, then

$$[T_r][T_{r-1}] \cdots [T_2][T_1][A]\{x\} = \lambda[T_r][T_{r-1}] \cdots [T_2][T_1]\{x\}. \tag{6}$$

Write $\{x\} = [T_1]^T [T_2]^T \cdots [T_{r-1}]^T [T_r]^T \{y\}$. Then

$$\begin{aligned}&[T_r][T_{r-1}] \cdots [T_2][T_1][A][T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T\{y\}\\ &\quad = \lambda[T_r][T_{r-1}] \cdots [T_2][T_1][T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T\{y\}\\ &\quad = \lambda\{y\},\end{aligned} \tag{7}$$

where Equation 3 is applied repeatedly to obtain the final form. It follows that $\{y\}$ is an eigenvector of the transformed matrix

$$[T_r][T_{r-1}] \cdots [T_2][T_1][A][T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T,$$

and that $\{x\}$ may be obtained from $\{y\}$ by

$$\{x\} = [T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T \{y\}. \tag{8}$$

The Givens method uses orthogonal matrices $[T]$ which are identical with the unit matrix $[I]$ except for the four elements

$$\left.\begin{aligned} t_{i+1,\, i+1} &= t_{j,\, j} = \cos\theta_{i+1,\, j} \, ,\\ t_{i+1,\, j} &= -t_{j,\, i+1} = \sin\theta_{i+1,\, j} \, .\end{aligned}\right\} \tag{9}$$

The orthogonal transformation $[T][A][T]^T$ leaves unchanged all the elements of $[A]$ except those in the $i+1^{st}$ and j^{th} rows and columns, the so-called plane of rotation. The four pivotal elements of the transformed matrix are

$$\left.\begin{aligned}
\alpha_{i+1,\,i+1} &= a_{i+1,\,i+1}\cos^2\theta_{i+1,\,j} + a_{j,\,j}\sin^2\theta_{i+1,\,j} + a_{i+1,\,j}\sin2\theta_{i+1,\,j}\,,\\
\alpha_{j,\,j} &= a_{i+1,\,i+1}\sin^2\theta_{i+1,\,j} + a_{j,\,j}\cos^2\theta_{i+1,\,j} - a_{i+1,\,j}\sin2\theta_{i+1,\,j}\,,\\
\alpha_{i+1,\,j} = \alpha_{j,\,i+1} &= a_{i+1,\,j}\cos2\theta_{i+1,\,j} - \frac{1}{2}\,(a_{i+1,\,i+1} - a_{j,\,j})\sin2\theta_{i+1,\,j}\,.
\end{aligned}\right\} \qquad (10)$$

where $a_{j,\,j}$ etc., are elements of the untransformed matrix. The other elements of the $i+1^{st}$ and j^{th} rows and columns of the transformed matrix are:

$$\left.\begin{aligned}
\alpha_{i+1,\,s} = \alpha_{s,\,i+1} &= a_{i+1,\,s}\cos\theta_{i+1,\,j} + a_{j,\,s}\sin\theta_{i+1,\,j}\\
\alpha_{j,\,s} = \alpha_{s,\,j} &= -a_{i+1,\,s}\sin\theta_{i+1,\,j} + a_{j,\,s}\cos\theta_{i+1,\,j}
\end{aligned}\right\} \qquad (11)$$

In the Givens method $\theta_{i+1,\,j}$ is chosen such that $\alpha_{i,\,j}$ will vanish, which happens when

$$\tan\theta_{i+1,\,j} = a_{i,\,j}/a_{i+1,\,i}. \qquad (12)$$

The calculation of $\theta_{i+1,\,j}$ followed by the orthogonal transformation

$$[A^{(m)}] = [T_m][A^{(m-1)}][T_m]^T, \qquad (13)$$

is carried out for a sequence of iterations with $[A^{(0)}] = [A]$. The values of i used in Equations 9, 10, 11, and 12 are 1, 2, 3, ..., (n-2) and for each i, a set of n - i - 1 such transformations is performed, with j taking values i + 2, i + 3, ..., n before the next value of i is used. The result is that elements in the matrix positions (1,3), (1,4), ..., (1,n), (2,4), (2,5), ..., (2,n), ..., (n-2,n) are successively reduced to zero, together with their transposes, the (3,1), (4,1), ..., (n,n-2) elements. The set of transformations thus reduces the matrix to tridiagonal form.

NASTRAN employs a procedure introduced by Wilkinson (Reference 4) in which the Givens method is modified by grouping together the (n - i - 1) transformations which produce zeros in the i^{th} row and column. This procedure should not be confused with the Householder method, Reference 3, which annihilates a row and column at a time. The Wilkinson process is particularly advantageous when the matrix [A] is so large that the elements cannot all be left in core storage at one time; it requires only n - 2 transfers of the matrix to and from auxiliary storage instead of the (n-1)(n-2)/2 transfers required by the unmodified Givens method. The method requires 4n core storage locations for working space and these are divided into four groups of n storage locations each. The first (i-1) rows and columns play no part in the ith major step. This step has five substeps:

1. The ith row of [A] is transferred to the set of core storage locations in group 1.

2. The values of $\cos\theta_{i+1,\,i+2}$, $\sin\theta_{i+1,\,i+2}$ .... $\cos\theta_{i+1,\,n}$, $\sin\theta_{i+1,\,n}$ are computed successively from

$$\left.\begin{aligned} \cos\theta_{i+1,\,j} &= \frac{a_{i,\,i+1}^{(j-1)}}{\sqrt{\left(a_{i,\,i+1}^{(j-1)}\right)^2 + a_{i,\,j}^2}}\,, \\ \sin\theta_{i+1,\,j} &= \frac{a_{i,\,j}}{\sqrt{\left(a_{i,\,i+1}^{(j-1)}\right)^2 + a_{i,\,j}^2}}\,, \end{aligned}\right\} \tag{14}$$

where the superscripted term is computed by

$$a_{i,\,i+1}^{(j-1)} = \sqrt{\left(a_{i,\,i+1}^{(j-2)}\right)^2 + a_{i,\,j-1}^2}\,,$$

and the starting value for j = i + 2 is

$$a_{i,\,i+1}^{(i+1)} = a_{i,\,i+1}\,. \tag{15}$$

The $\cos\theta_{i+1,j}$ may be overwritten on the elements of the untransformed matrix, $a_{i,\,j}$, which are no longer required, and the $\sin\theta_{i+1,j}$ are stored in the group 2 storage locations.

3. The i+1st row of [A] is transferred to the group 3 storage locations. Only those elements on and above the diagonal are used in this and succeeding rows.

For k = i + 2, i + 3, ..., n in turn, the operations in substeps 4 and 5 are carried out.

4. The kth row of [A] is transferred to the group 4 storage locations. The elements $a_{i+1,\,i+1}$, $a_{i+1,\,k}$ and $a_{k,\,k}$ are subjected to the row and column operations involving $\cos\theta_{i+1,k}$ and $\sin\theta_{i+1,k}$. For $\ell = k + 1, k + 2, \ldots, n$ in turn, the part of the row transformation involving $\cos\theta_{i+1,\,k}$ and $\sin\theta_{i+1,\,k}$ is performed on $a_{i+1,\,\ell}$ and $a_{k,\,\ell}$. For $\ell = k + 1, k + 2, \ldots, n$ in turn, the part of the column transformation involving $\cos\theta_{i+1,\,\ell}$ and $\sin\theta_{i+1,\,\ell}$ is performed on $a_{i+1,\,k}$ and $a_{k,\,\ell}$. Now all the transformations involving the ith major step have been performed on all the elements of row k and on elements i + 2, i + 3, ..., n of row i + 1.

5. The completed k^{th} row is transferred to auxiliary storage.

When substeps 4 and 5 have been completed for all appropriate values of k, the work on row i + 1 has also been completed. The values of $\cos\theta_{i+1,\,k}$ and $\sin\theta_{i+1,\,k}$ for $k = i + 2, i + 3, \ldots, n$ are transferred to auxiliary storage and row i + 1 is transferred to the group 1 core storage locations. Since the $i + 1^{st}$ row plays the same part in the next major step as did the i^{th} in the step just described, everything is ready for substep 2 in the next major step. Substep 1 is, in fact, only required in the first major step because the appropriate row is already in the storage locations of group 1 in subsequent steps.

10.2.3 Extraction of the Eigenvalues of a Tridiagonal Matrix

NASTRAN employs the Q-R transformation of Francis (Reference 5) which is an orthogonal transformation $[A_{r+1}] = [Q_r]^T [A_r] [Q_r]$ such that $[A_r]$ may be factored into the product $[Q_r] [R_r]$ where $[R_r]$ is an upper triangular matrix. Thus

$$[A_r] = [Q_r][R_r] \; , \tag{16}$$

and

$$\begin{aligned}[A_{r+1}] &= [Q_r]^T[A_r][Q_r] \\ &= [Q_r]^T[Q_r][R_r][Q_r] \; .\end{aligned}$$

Now $[Q_r]^T[Q_r] = [I]$ by virtue of the orthogonality property, Equation 3, so that

$$[A_{r+1}] = [R_r][Q_r] \; . \tag{17}$$

It follows that $[A_{r+1}]$ is determined from $[A_r]$ by performing in succession the decomposition given by Equation 16 and the multiplication given by Equation 17. Francis has shown that if a matrix $[A] = [A_1]$ is nonsingular, then, in the limit as $r \to \infty$, $[A_r]$ will approach an upper triangular matrix; because eigenvalues are preserved under orthogonal transformation, it follows that diagonal elements of the limiting matrix are the eigenvalues of the original matrix [A]. Although the method can be applied to any matrix, it is particularly suitable for tridiagonal matrices because the bandwidth of the matrix can be preserved, as will be shown. In the case where [A] is symmetric, the matrix $[A_r]$ will, of course, tend to a diagonal form as $r \to \infty$.

It may be shown that the upper triangular matrix, $[R_r]$, and the orthogonal matrix $[Q_r]$ are unique, Reference 4, page 241; several methods, however, are available for performing the decomposition. In the method of calculation devised by Francis, $[Q_r]$ is expressed as a product of (n-1) elementary rotation matrices, where n is the order of $[A_r]$:

$$[Q_r] = [T^{(1)}][T^{(2)}] \dots [T^{(n-1)}] \; . \tag{18}$$

The nonzero elements of the j^{th} elementary rotation matrix are

$$\begin{aligned} t^{(j)}_{j,\,j} &= t^{(j)}_{j+1,\,j+1} = c_j \; , \\ t^{(j)}_{j+1,\,j} &= -t^{(j)}_{j,\,j+1} = s_j \; , \\ t^{(j)}_{k,\,k} &= 1, \; k \neq j \text{ or } j+1. \end{aligned} \tag{19}$$

The manner in which the c_j and s_j coefficients are obtained from the elements of $[A_r]$ will be shown shortly. $[R_r]$ is, from Equation 16 and the orthogonality property,

$$\begin{aligned} [R_r] &= [Q_r]^{-1}[A_r] = [Q_r]^T[A_r] \\ &= [T^{(n-1)}]^T[T^{(n-2)}]^T \dots [T^{(2)}]^T[T^{(1)}]^T[A_r] \; . \end{aligned} \tag{20}$$

Let the nonzero elements of $[A_r]$, $[A_{r+1}]$ and $[R_r]$ be defined as follows:

$$[A_r] = \begin{bmatrix} a_1 & b_2 & & & 0 \\ b_2 & a_2 & b_3 & & \\ & \cdot & \cdot & \cdot & \\ 0 & & b_{n-1} & a_{n-1} & b_n \\ & & & b_n & a_n \end{bmatrix} \tag{21}$$

$$[A_{r+1}] = \begin{bmatrix} \bar{a}_1 & \bar{b}_2 & & & 0 \\ \bar{b}_2 & \bar{a}_2 & \bar{b}_3 & & \\ & \cdot & \cdot & \cdot & \\ & & \bar{b}_{n-1} & \bar{a}_{n-1} & \bar{b}_n \\ 0 & & & \bar{b}_n & \bar{a}_n \end{bmatrix} , \qquad (22)$$

and

$$[R_r] = \begin{bmatrix} r_1 & q_1 & t_1 & & 0 \\ & r_2 & q_2 & t_2 & \\ & & \cdot & \cdot & \cdot \\ & 0 & & r_{n-1} & q_{n-1} \\ & & & & r_n \end{bmatrix} . \qquad (23)$$

The coefficients of the elementary rotation matrices are selected so as to reduce the subdiagonal terms of $[R_r]$ to zero. Specifically,

$$\left. \begin{aligned} s_j &= \frac{b_{j+1}}{\left(p_j^2 + b_{j+1}^2\right)^{\frac{1}{2}}} , \\ c_j &= \frac{p_j}{\left(p_j^2 + b_{j+1}^2\right)^{\frac{1}{2}}} , \end{aligned} \right\} \quad j = 1, 2, \ldots, n-1 , \qquad (24)$$

where

$$\left. \begin{aligned} p_1 &= a_1 , \\ p_2 &= c_1 a_2 - s_1 b_2 , \\ p_j &= c_{j-1}\, a_j - s_{j-1}\, c_{j-2}\, b_j . \qquad j = 3, 4, \ldots, n-1 . \end{aligned} \right\} \qquad (25)$$

Substitution into Equation 20 yields the elements of $[R_r]$.

$$\left.\begin{aligned}
&r_j = c_j p_j + s_j b_{j+1} , && j = 1, 2, \ldots, n-1 , \\
&r_n = p_n , && \\
&q_1 = c_1 b_2 + s_1 a_2 , && \\
&q_j = s_j a_{j+1} + c_j c_{j-1} b_{j+1} , && j = 2, 3, \ldots, n-1 , \\
&t_j = s_j b_{j+2} , && j = 1, 2, \ldots, n-2 .
\end{aligned}\right\} \quad (26)$$

The elements of $[A_{r+1}]$ are, from Equation 17,

$$\left.\begin{aligned}
&\bar{a}_1 = c_1 r_1 + s_1 q_1 , && \\
&\bar{a}_j = c_{j-1} c_j r_j + s_j q_j , && j = 2, 3, \ldots, n-1 , \\
&\bar{a}_n = c_{n-1} r_n , && \\
&\bar{b}_{j+1} = s_j r_{j+1} , && j = 1, 2, \ldots, n-1 .
\end{aligned}\right\} \quad (27)$$

NASTRAN uses a variation, Reference 6, of Francis' original method that avoids the calculation of square roots. This is done by using the following equations in place of Equation 27

$$\left.\bar{a}_j = (1 + s_j^2)\, g_j + s_j^2\, a_{j+1} , \qquad j = 1, 2, \ldots, n-1 \right\} \quad (28)$$

$$\left.\begin{aligned}
&\bar{a}_n = g_n , && \\
&\bar{b}_{j+1}^2 = s_j^2 (p_{j+1}^2 + b_{j+2}^2) , && j = 1, 2, \ldots, n-2 , \\
&\bar{b}_n^2 = s_{n-1}^2\, p_n^2 , &&
\end{aligned}\right\} \quad (29)$$

Where,

$$\left.\begin{aligned}
g_1 &= a_1 \, , \\
g_j &= c_{j-1}\, p_j = a_j - s_{j-1}^2\,(a_j + g_{j-1}) \, , && j = 2, 3, \ldots, n \, , \\
p_1^2 &= a_1^2 \, , \\
p_j^2 &= g_j^2 \,/\, c_{j-1}^2 \quad \text{if} \quad c_{j-1} \neq 0 \, , && j = 2, 3, \ldots, n \, , \\
&= c_{j-2}^2\, b_j^2 \quad \text{if} \quad c_{j-1} = 0 \, , && j = 2, 3, \ldots, n \, .
\end{aligned}\right\} \qquad (30)$$

The reason that the use of Equations 28, 29 and 30 in place of Equation 27 avoids the calculation of square roots can best be seen by considering the terms input to and produced by these equations.

For Equation 27, the input terms consist of the elements of $[A_r]$. That is; $a_1, a_2, \ldots, a_n$ and $b_2, b_3, \ldots, b_n$. The data produced is $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_n$ and $\bar{b}_2, \bar{b}_3, \ldots, \bar{b}_n$ which are the elements of $[A_{r+1}]$. Hence, this completes one iteration step, but involves the calculation of square roots.

However, for Equations 28, 29, and 30, the input terms consist of $a_1, a_2, \ldots, a_n$ and $b_1^2, b_2^2, \ldots, b_n^2$. The data produced are $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_n$ and $\bar{b}_1^2, \bar{b}_2^2, \ldots, \bar{b}_n^2$ which serve as the input to the next iteration step. Here no square roots need to be computed.

Convergence of the tridiagonal matrix to a diagonal form is speeded up by origin shifting. NASTRAN employs a procedure suggested by Wilkinson (Reference 7), who has shown that when the eigenvalues are real, the best shift strategy is to subtract a_n from each diagonal element of $[A]$, thereby reducing each eigenvalue by a_n.

Another useful device in speeding up the determination of eigenvalues of tridiagonal matrices takes advantage of zeroes which may occur in the off-diagonal rows. Let the $[A_r]$ matrix be represented in partitioned form as follows.

$$\left[\begin{array}{c|c|c}
\begin{matrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & & \ddots & & \\ & & & a_{j-2} & b_{j-2} \\ & & & b_{j-2} & a_{j-1} \end{matrix} & & \\ \hline
 & \begin{matrix} a_j & b_j & & & \\ b_j & a_{j+1} & b_{j+1} & & \\ & & \ddots & & \\ & & b_{m-2} & a_{m-1} & b_{m-1} \\ & & & b_{m-1} & a_m \end{matrix} & \\ \hline
 & & \begin{matrix} a_{m+1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_n \end{matrix}
\end{array}\right] \tag{31}$$

In this matrix the diagonal terms in the lower right hand partition are eigenvalues that have been determined in previous iterations. The j^{th} row is the next lowest in which the off-diagonal term, b_{j-1}, vanishes, thereby uncoupling the equations in the first j-1 rows from the rest. As a result, the eigenvalues of the matrix in the central block may be obtained separately. Other uncoupled blocks may be found in the upper left partition.

The iteration described by Equations 28, 29 and 30 is continued until $\bar{b}_{m-1}^2$ vanishes to satisfactory accuracy so that $\bar{a}_m$ may be accepted as an eigenvalue of the shifted matrix. $\bar{b}_{m-1}^2$ must be negligible compared to $\bar{a}_m^2$. ($\bar{b}_{m-1}^2 + \bar{a}_m^2$ must approximately equal $\bar{a}_m^2$). $\bar{a}_m$ is transferred to the lower partition and the process is continued until all the eigenvalues of the partitioned matrix have been extracted.

10.2.4 Computation of Eigenvectors

The eigenvector corresponding to any eigenvalue, λ_i, of the tridiagonal matrix may be determined by solving (n-1) of the equations:

$$\left.\begin{aligned} (a_1 - \lambda_i)\, x_1 + b_2 x_2 &= 0 \ , \\ b_2\, x_1 + (a_2 - \lambda_i)\, x_2 + b_3 x_3 &= 0 \ , \\ &\ \vdots \\ b_{n-1}\, x_{n-1} + (a_{n-1} - \lambda_i)\, x_{n-1} + b_n x_n &= 0 \ , \\ b_n\, x_{n-1} + (a_n - \lambda_i)\, x_n &= 0 \ . \end{aligned}\right\} \tag{32}$$

If the first (n - 1) equations are used, the solution is obtained by taking x_1 to be unity and back substituting in the equations to obtain values of $x_2, x_3, \ldots, x_n$. Wilkinson (Reference 7) has shown that this method is unstable and has suggested the following approach which is used in NASTRAN.

The tridiagonal matrix $[A - \lambda_i I]$ is factored into the product of a lower unit triangle $[L_i]$ and an upper triangle $[U_i]$. In the decomposition partial pivoting is used, i.e., the pivotal row at each stage is selected to be that equation which has the largest coefficient of the variable being eliminated. At each stage there will be only two equations containing that variable. The eigenvector $\{\phi_i\}$ is then obtained from solution of the equation

$$[U_i]\{\phi_i\} = \{C\} \quad , \tag{33}$$

where {C} is arbitrarily selected. The solution is easily obtained by back substitution because $[U_i]$ has the form

$$[U_i] = \begin{bmatrix} p_1 & q_1 & r_1 & & & \\ & p_2 & q_2 & r_2 & & \\ & & & \ddots & & \\ & & & p_{n-2} & q_{n-2} & r_{n-2} \\ & & & & p_{n-1} & q_{n-1} \\ & & & & & p_n \end{bmatrix} . \tag{34}$$

An improved solution is obtained by repeated application of Equation 33 using the current estimate of $\{\phi_i\}$ on the right hand side. Thus

$$[U_i]\{\phi_i^{(n)}\} = \{\phi_i^{(n-1)}\} \; , \tag{35}$$

where $\{\phi_i^{(0)}\} = \{C\}$. Wilkinson, Reference 7, shows that, if the computed eigenvalue λ_i is a close approximation to the true eigenvalue, convergence is so rapid that more than two iterations are never required. The test applied is that the maximum component of the eigenvector does not change (to single precision accuracy) in one iteration. The initial vector {C} is chosen so that each element is unity.

In the case of a double eigenvalue, the above method gives one eigenvector $\{\phi_1\}$. If we start with any initial vector {b} orthogonal to $\{\phi_1\}$ and apply the previous algorithm, convergence to $\{\phi_2\}$, the other eigenvector, will result. The following procedure is used in NASTRAN to extract multiple eigenvalues. If eigenvectors $\{\phi_1\}$, $\{\phi_2\}$, ..., $\{\phi_m\}$ with elements $\{\phi_s\} = \lfloor a_{1s}, a_{2s}, \ldots, a_{ns}\rfloor^T$ have been obtained, an initial vector {b} orthogonal to each of these eigenvectors is obtained by taking the components $b_{m+1}, b_{m+2}, \ldots, b_n$ as unity and solving the simultaneous equations

$$\left.\begin{array}{l}
b_1\, a_{11} + b_2\, a_{21} + \ldots + b_m\, a_{m1} = -\displaystyle\sum_{s=m+1}^{n} (a_{s1}) \; , \\[2ex]
b_1\, a_{12} + b_2\, a_{22} + \ldots + b_m\, a_{m2} = -\displaystyle\sum_{s=m+1}^{n} (a_{s2}) \; , \\
\quad\cdot \\
\quad\cdot \\
\quad\cdot \\
b_1\, a_{1m} + b_{2m}\, a_{2m} + \ldots + b_m a_{mm} = -\displaystyle\sum_{s=m+1}^{n} (a_{sm}) \; .
\end{array}\right\} \tag{36}$$

Accumulated rounding errors will result in the computed multiple eigenvectors not being exactly orthogonal to one another. The following Gram-Schmidt algorithm described by Bodewig (Reference 8) is used to produce an orthogonal set of k eigenvectors $\{y_s\}$ from the almost orthogonal set $\{x_s\}$. For s = 1, select

$$\{y_1\} = \{x_1\} \,/\, \|\, \{x_1\}\, \| \; . \tag{37}$$

Then for $1 < s \leq k$, calculate

ORIGINAL PAGE IS
OF POOR QUALITY

$$\left.\begin{aligned} \{z_s\} &= \{x_s\} - \sum_{t=1}^{s-1} (\{x_s\}^T \{y_t\}) \{y_t\} \text{ , and} \\ \{y_s\} &= \{z_s\} / \| \{z_s\} \| \text{ ,} \end{aligned}\right\} \quad (38)$$

where $\|\{z_s\}\|$ denoted the Euclidean norm, $\sqrt{z_{s1}^2 + z_{s2}^2 + \ldots + z_{sn}^2}$, of a vector $\lfloor z_{s1}, z_{s2}, \ldots, z_{sn} \rfloor^T$, and where $\{x_s\}^T \{y_t\}$ is a scalar product of the vectors $\{x_s\}$ and $\{y_t\}$.

Finally, when all the eigenvectors of the tridiagonal matrix have been extracted, the matrix multiplications of Equation 8 are carried out to obtain the eigenvectors of the original matrix [A].

EIGENVALUE EXTRACTION METHODS

REFERENCES

1. Givens, W., "Numerical Computation of the Characteristic Values of a Real Symmetric Matrix," Oak Ridge National Lab., ORNL-1574, 1954.
2. Lanczos, C., "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators," J. Res. Nat. Bur. Stand., Vol. 45, 1950, pp. 255-282.
3. Householder, A. S., and Bauer, F. L., "On Certain Methods for Expanding the Characteristic Polynomial," Numerische Methematik, Vol. 1, 1959, pp. 29-37.
4. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965, pp. 506-510.
5. Francis, J. G. F., "The QR Transformation, a Unitary Analogue to the LR Transformation," The Computer Journal, Vol. 4, No. 3 (Oct. 1961) and No. 4 (Jan. 1962).
6. Ortega, J. M. and Kaiser, H. F., "The LL^T and QR Methods for Symmetric Tridiagonal Matrices," The Computer Journal, Vol. 6, No. 1, January 1963, pp. 99-101.
7. Wilkinson, J. H., "The Calculation of the Eigenvectors of Codiagonal Matrices," The Computer Journal, Vol. I, 1958, p. 90.
8. Bodewig, E., Matrix Calculus, Interscience Publishers, 1959.

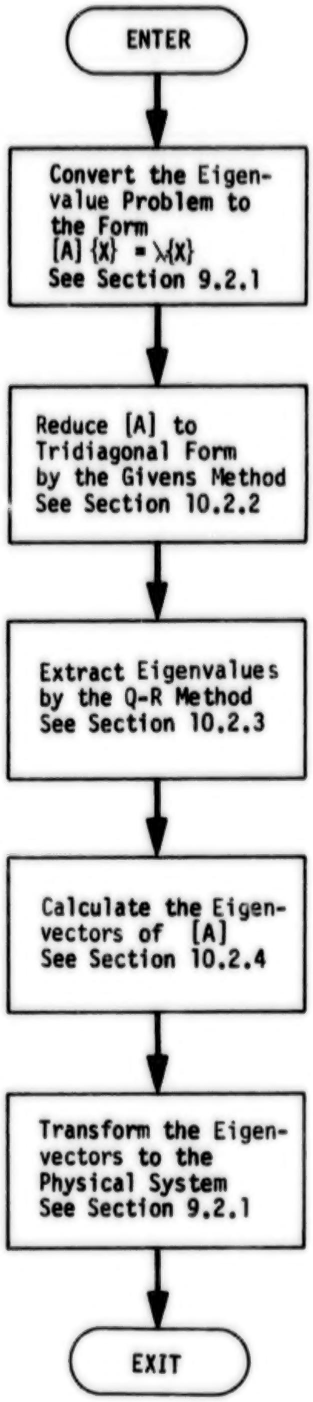

ENTER
Convert the Eigenvalue Problem to the Form [A] {X} = λ{X} See Section 9.2.1
Reduce [A] to Tridiagonal Form by the Givens Method See Section 10.2.2
Extract Eigenvalues by the Q-R Method See Section 10.2.3
Calculate the Eigenvectors of [A] See Section 10.2.4
Transform the Eigenvectors to the Physical System See Section 9.2.1
EXIT

Figure 1. Simplified flow diagram for the tridiagonal method.

10.3 THE DETERMINANT METHOD

10.3.1 Fundamentals of the Determinant Method

The basic notion employed in the determinant method of eigenvalue extraction is very simple. If the elements in a matrix [A] are polynomial functions of the operator p, then the determinant of [A] can be expressed as

$$D(A) = (p - p_1)(p - p_2)(p - p_3)\ldots(p - p_n) \quad , \qquad (1)$$

where $p_1, p_2, p_3 \ldots p_n$ are the eigenvalues of the matrix. The value of the determinant vanishes for $p = p_i$, $i = 1,2,3 \ldots n$.

In the determinant method, the determinant is evaluated for trial values of p, selected according to some iterative procedure, and a criterion is established to determine when D(A) is sufficiently small or when p is sufficiently close to an eigenvalue. The eigenvector is then found by solution of the equation

$$[A]\{u\} = 0 \quad , \qquad (2)$$

with one of the elements of {u} preset.

The most convenient procedure for evaluating the determinant of a matrix employs the triangular decomposition

$$[A] = [L][U] \quad , \qquad (3)$$

where [L] is a lower unit triangular matrix (unit values on the diagonal) and [U] is an upper triangular matrix. The determinant of [A] is equal to the product of the diagonal terms of [U].

Two versions of the triangular decomposition operation are provided. In the standard version row interchanges are used in order to improve numerical stability. In an optional version, which is available for real eigenvalue extraction only, row interchanges are not used. The optional version is approximately four times as fast as the standard version for banded matrices, but it includes a small risk of numerical failure due to the fact that [A] is seldom a positive definite matrix. The algorithms employed in both versions are described in Section 2.2.

The matrix [A] may be expressed as

$$[A] = -p[M] + [K] \quad , \qquad (4)$$

PAGE 411 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

412

for real eigenvalue problems, and as

$$[A] = p^2[M] + p[B] + [K] \quad , \tag{5}$$

for complex eigenvalue problems.

The determinant method is not a particularly efficient method because a large number of triangular decompositions of the [A] matrix are performed if more than a few eigenvalues are desired. An exception occurs in the case of very narrowly banded matrices, such as those in beam problems, where matrix decomposition is a relatively rapid operation. The main strength of the determinant method is its insensitivity to the functional form of the elements of the [A] matrix which could, for example, contain poles as well as zeroes, or be transcendental functions of p. These effects occur frequently in hydroelastic and aeroelastic problems.

10.3.2 Iteration Algorithm

Wilkinson's recent, but already standard, treatise (Reference 1) includes an authoritative discussion of polynomial curve-fitting schemes for tracking the roots of a determinant. He shows that little is to be gained by using polynomials higher than the second degree. Accordingly, Muller's quadratic method (Reference 1, page 435) is used in NASTRAN. The form of the algorithm is as follows.

Consider a series of determinants, D_{k-2}, D_{k-1}, D_k, evaluated for trial values of the eigenvalue, $p = p_{k-2}, p_{k-1}, p_k$. A better approximation to the eigenvalue is obtained from the following calculations. Let

$$h_k = p_k - p_{k-1} \quad , \tag{6}$$

$$\lambda_k = \frac{h_k}{h_{k-1}} \quad , \tag{7}$$

$$\delta_k = 1 + \lambda_k \quad . \tag{8}$$

Then

$$h_{k+1} = \lambda_{k+1} h_k \quad , \tag{9}$$

$$p_{k+1} = p_k + h_{k+1} \quad , \tag{10}$$

where

$$\lambda_{k+1} = \frac{-2D_k\delta_k}{g_k \pm \left[g_k^2 - 4D_k\delta_k\lambda_k\left(D_{k-2}\lambda_k - D_{k-1}\delta_k + D_k\right)\right]^{1/2}} \quad , \tag{11}$$

in which

$$g_k = D_{k-2}\lambda_k^2 - D_{k-1}\delta_k^2 + D_k(\lambda_k + \delta_k) \quad . \tag{12}$$

The (+) or (-) sign in Equation (11) is selected to minimize the absolute value of λ_{k+1}. In the case when p_k, p_{k-1}, and p_{k-2} are all arbitrarily selected initial values (starting points), the starting points are arranged such that

$$D_k \leqslant D_{k-1} \leqslant D_{k-2} \quad , \tag{13}$$

and the (+) or (-) sign in Equation 11 is selected to minimize the distance from p_{k+1} to the closest starting point rather than to p_k.

In a real eigenvalue analysis, it is possible to calculate a complex value of λ_{k+1} from Equation 11. In order to preclude the occurrence of complex arithmetic in a real eigenvalue analysis, only the real part of λ_{k+1} is used to estimate p_{k+1}. The real part corresponds to the minimum absolute value of the parabolic approximation.

10.3.3 Scaling

In calculating the determinant of [A] (Equation 3), some form of scaling must be employed, because the accumulated product will rapidly overflow or underflow the floating point size of a digital computer. Accordingly, the accumulated product of the diagonal terms of [U] is calculated and stored as a scaled number

$$D = d \times 10^n \quad , \tag{14}$$

where

$$\frac{1}{10} \leq d < 1 \quad . \tag{15}$$

The arithmetic operations indicated in Equations 11 and 12 are calculated in scaled arithmetic. λ_{k+1} is then reverted to unscaled form.

10.3.4 Sweeping of Previously Extracted Eigenvalues

Once an eigenvalue has been found to satisfactory accuracy, a return to that eigenvalue by the iteration algorithm can be prevented by dividing the determinant by the factor $(p - p_i')$, where p_i' is the accepted approximation to p_i, in all subsequent calculations.

Thus

$$D^{(1)}(A) = \frac{D(A)}{p - p_1'} \quad , \tag{16}$$

should be used in place of D(A) after the first eigenvalue has been found. In general, the reduced determinant used for finding the i+1st eigenvalue is

$$D^{(i)}(A) = \frac{D^{(i-1)}(A)}{p - p_i'} = \frac{D(A)}{(p - p_1')(p - p_2') \; . \; . \; . \; (p - p_i')} \quad . \tag{17}$$

Wilkinson states that this sweeping procedure is quite satisfactory provided that all p_i' have been calculated to an accuracy that is limited only by round-off error.

In some instances there are known eigenvalues for which calculations need not be made. An important example is the set of rigid body (zero-frequency) modes that are calculated by a separate subroutine (see Section 9.2). In addition, the user may know of other eigenvalues, such as those extracted in a previous execution or those resulting from transfer functions (see Section 9.3), that should be avoided. A special data card (EIGP) is used in complex analysis to specify the location of such roots. They are then eliminated from the determinant by a preliminary operation

$$D^{(0)}(A) = \frac{D(A)}{\prod_k (p - p_k)^{m_k}} \quad , \tag{18}$$

where m_k is the multiplicity of the known eigenvalue, p_k.

For problems with conjugate complex eigenvalues (complex eigenvalue analysis with real matrices) the conjugates of the extracted eigenvalues are also swept from the determinant. Thus

$$D^{(i)}(A) = \frac{D^{(i-1)}(A)}{(p - p_i')(p - \bar{p}_i')} \quad , \tag{19}$$

where $\bar{p}_i'$ is the conjugate of p_i'.

There is a danger in this procedure that if p_i' is very near the axis of reals, the exact eigenvalue may, in fact, be real. To avoid this situation a test is applied to the imaginary part of p_i' and the conjugate of p_i' is swept only if

$$|\mathrm{Im}\ p_i'| \geqslant 1000.0(R_{max})(\epsilon) \quad . \tag{20}$$

The parameters R_{max} and ϵ are discussed in the next two sections.

10.3.5 Search Procedures

Three initial values of p (starting points) are needed in order to start the iteration algorithm. The determinant method is essentially a root-tracking method that finds nearby roots with ease and remote roots with difficulty. Thus it is not advisable to use the same three starting points for all eigenvalues because the eigenvalues are usually distributed throughout a region of the p-plane and at lease some of the eigenvalues will be remote from any given set of three starting points.

In the case of real eigenvalue analysis, the starting points are initially uniformly distributed in an interval of p. The user specifies the lowest and highest eigenvalues of interest, R_{min} and R_{max}, and also supplies an estimate, N_E, of the number of roots in the range. A total of $2N_E + 1$ starting points are then initially located such that the first coincides with R_{min} and the rest are placed at uniform intervals between R_{min} and R_{max}. It has, however, been found that this distribution of starting points, while satisfactory in most cases, causes computational problems when the higher frequency in any interval is larger than the lower frequency in that interval by several orders of magnitude. In order to avoid this difficulty, any initial interval in which the higher frequency is larger than (or equal to) ten times the lower frequency in that range is further subdivided into smaller unequal intervals, resulting in additional unequally spaced (as per a geometrical progression) starting points. This scheme thus guarantees that every starting point has a frequency that is less than ten times the frequency of the next lower starting point. Accordingly, the resulting number of starting points may therefore be larger than $2N_E + 1$.

The process of extracting eigenvalues is initiated from the three smallest starting points, p_{s1}, p_{s2}, and p_{s3}. After all eigenvalues within the "range" of these starting points have been extracted, p_{s1} is dropped, p_{s4} is added, and the search is repeated. The details will be described later.

The search procedure for complex eigenvalues is more complicated because the roots are distributed throughout a plane rather than along a line. Fortunately, in structural analysis, most roots are found near the imaginary axis so that a set of starting points placed along the imaginary axis will usually produce satisfactory results. In order to provide for more general problems, rectangular search regions may be located by the user in any part of the complex plane as shown in Figure 1. It is intended that all eigenvalues within the search regions be extracted (as limited by the desired maximum number of roots). There may be any number of search regions, and the search regions may overlap. The search regions are established by means of the coordinates of end points (A_j, B_j) and the width of the region (ℓ_j). The user estimates the number of roots

(N_{ej}) in each search region. For problems with real coefficients, all roots are either real or they occur in conjugate complex pairs, and it is inefficient for the user to specify regions in the lower half of the complex plane since the existence of conjugate pairs is assumed. The user will not, however, be prevented from specifying a search in the lower half of the complex plane since it is useful, in many cases, for a search region to include the origin or a band along the real axis.

The steps in the search procedure for complex eigenvalue extraction are as follows:

1. Select $2N_{ej} + 2$ starting points equally distributed along the line A_j, B_j. Do not use A_j and B_j as starting points.
2. Find the starting point in Region I that is nearest to the origin as shown in Figure 2 and designate it p_{s1}. Draw a line perpendicular to A_1B_1 midway between p_{s1} and p_{s1}', the point next nearest to the origin, thereby dividing Region I into Regions IA and IB.
3. Select the three starting points in Region IA nearest to the line, a - a', as the initial set and proceed to extract roots. Then proceed to the three starting points in Region IB (p_{s1}', p_{s2}', p_{s3}') that are closest to the line a - a'. Then return to the points p_{s2}, p_{s3}, p_{s4} in Region IA, and, in general, alternate back and forth between the two regions until all starting points have been used once, or until termination occurs for some other reason.
4. When all starting points in Region I have been used once, proceed to Region II, etc. Sweep out all previously extracted roots from whatever region, in evaluating determinants.
5. When all starting points in all regions have been used once, return to Region I, II, etc. in turn and repeat the above procedure. Continue to repeat until no new roots are found in any pass through the regions or until termination occurs for some other reason.

When searching for either real or complex roots from any set of starting points, the search is terminated if a root is predicted to lie outside of the local search region. In the case of complex analysis, the local search region for the current starting point set is bounded by lines parallel to A_jB_j at a distance $\frac{\ell}{2}$ and extends from the first starting point on the other side of a - a' to a line that is 55% of the way between the current starting point that is farthest from a - a' and the starting point that will be picked up in the next set (see Figure 2).

PAGE 10.3-5B INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

649

In the case of real eigenvalue analysis the local search region extends from R_{min} to a point that is 55% of the way from the largest current starting point to the one that will be picked up in the next set. The reason for permitting the local search region to extend all the way back to R_{min} is to improve the chance of finding eigenvalues that were initially bypassed. Additional assurance is obtained by making repeated passes through all of the starting points until no new eigenvalues are found on one complete pass. The computational effort for the last pass is trivial because the determinants for all starting points are saved.

The search for eigenvalues is finally terminated when no additional roots are found in one complete pass through all regions or when the maximum desired number of roots for each region, N_{dj}, as requested by the user, have been extracted.

Failure to find additional roots normally occurs because all of the roots within the desired region(s) have been extracted. Situations can occur in which some roots will be missed. The most common occurs in complex eigenvalue analysis wher one or more desired roots lie at a large distance from the centerline of the region, see Figure 2, and several other roots lie just beyond the ends of the region. The search procedure is likely to be attracted toward the latter roots and may not find the former. The possibility of such failures is one reason for permitting multiple search regions. Another possible reason for missing roots is failure of the iteration algorithm to converge, as discussed below.

10.3.6 Convergence Criteria

The convergence criteria are based on successive values of the increment, h_k, in the estimated eigenvalue. No tests on the magnitude of the determinant or on any of the diagonal terms of the triangular decomposition are necessary or desirable.

Wilkinson (Reference 1, page 437) shows that, for h_k sufficiently small, the magnitude of h_k is approximately squared for each successive iteration when using Muller's method to find isolated roots. This is an extremely rapid rate of convergence. In a very few iterations, the "zone of indeterminacy" is reached within which h_k remains small, but exhibits random behavior due to round-off error.

Wilkinson states that, if it is desired to calculate the root to the greatest possible precision, the convergence criterion for accepting p_k as a root should be

$$|h_{k+1}| > |h_k| \tag{21}$$

We accept his advice, tempered by practical considerations. The first of these is that Equation 21 may be satisfied during the first few iterations while the root tracking algorithm is picking up the "scent". Thus it must, in addition, be required that $|h_k|$, $|h_{k-1}|$ and $|h_{k-2}|$ be reasonably small. The second practical consideration is that several iterations may be wasted within the zone of indeterminacy while waiting for Equation 21 to be satisfied. This can be avoided by accepting p_k if $|h_k|$ is sufficiently small. Finally, if the number of iterations becomes excessively large without satisfying a convergence criterion, it is best to give up and proceed to a new set of starting points.

Figure 3 is a flow diagram of a set of tests which meet the requirements discussed above for real eigenvalue problems. The tests are based on calculated values of $\overline{H}_1$, $\overline{H}_2$, and $\overline{H}_3$ which are defined as

$$\overline{H}_1 = \frac{|h_{k-1}|}{\sqrt{|p_k|}} \quad , \tag{22}$$

$$\overline{H}_2 = \frac{|h_k|}{\sqrt{|p_k|}} \quad , \tag{23}$$

$$\overline{H}_3 = \frac{|h_{k+1}|}{\sqrt{|p_k|}} \quad . \tag{24}$$

where p_k = kth estimate of an eigenvalue, and $h_k = p_k - p_{k-1}$.

The standard test in Figure 3 for accepting an eigenvalue is

$$\overline{H}_3 \leqslant 2\varepsilon \sqrt{R_{max}} \quad . \tag{25}$$

ε may be specified by the user. An equivalent expression, in terms of vibration mode frequencies, is

$$\frac{\omega_{k+1} - \omega_k}{\omega_{max}} \leqslant \varepsilon \quad , \tag{26}$$

where ω_{max} is the highest frequency of interest. A similar set of tests are performed for

651

complex eigenvalue problems.

The magnitude of the convergence criterion, ε, should be selected as a compromise between running time and accuracy. If, for example, it is found that the algorithm fails to find roots because of an excessive number of iterations, the value of ε should be increased. If failure occurs because the number of iterations exceeds the iteration limit, N_{IT}, for two successive sets of starting points, the value of ε is increased by a factor of ten. If successive pairs of failures still occur, ε is again increased by a factor of ten, until the number of permissible changes in ε, N_ε, is exceeded. The user is informed of the reduced precision of the calculations.

The values of the convergence parameters used in NASTRAN are

$$\varepsilon = 10^{-11} \quad ,$$

$$N_{IT} = 20 \quad ,$$

$$N_\varepsilon = 4 \quad .$$

The above value of ε may be overridden by the user.

The convergence rates for the Muller method are quite satisfactory for isolated roots and for double roots. For higher order multiple roots, the asymptotic rate of convergence is linear, i.e.,

$$h_{k+1} = a \cdot h_k \quad , \quad a < 1 \quad . \tag{27}$$

Convergence is progressively slower as the multiplicity increases. Zero frequency roots are the only kind that commonly occur with high multiplicity in structural analysis and provision is made for eliminating them, or other known roots, beforehand (see Section 10.3.4).

Experience with the determinant method indicates that, on average, convergence to isolated roots is achieved in about six iterations. The number of iterations increases for problems with multiple roots, many close roots, or roots clustered just beyond the ends of the desired region.

10.3.7 Test for Closeness to Starting Point

Once an eigenvalue has been extracted it is tested for closeness to one of the starting points, and if it is found to be too close, the starting point is shifted. The reason for the shift is that the value of the determinant near a root is small and contains considerable

652

round-off error, so that the value of the swept determinant, $D/(p - p_k)$ may have considerable error. The criterion for shifting in real eigenvalue analysis is

$$p_{s,i} - p_k < 400\varepsilon \sqrt{R_{max} p_k} \quad . \tag{28}$$

The magnitude of the shift is

$$(p_{s,i})_{shifted} = p_{s,i} + 2000\varepsilon \sqrt{R_{max} p_k} \quad . \tag{29}$$

10.3.8 Recovery of Eigenvectors

Once an approximate eigenvalue, p_j, has been accepted, the eigenvector is determined by back substitution into the previously computed triangular decomposition of $[A(p_j)]$. Now since

$$[A(p_j)]\,\{u\} = [L(p_j)][U(p_j)]\{u\} = 0 \quad , \tag{30}$$

and since $[L(p_j)]$ is nonsingular, only $[U(p_j)]$ is used. The last diagonal term in $[U(p_j)]$ is normally the only term with very small value. The normal appearance of $[U(p_j)]$ is as follows, for $n = 7$.

$$\begin{bmatrix} X & X & X & X & 0 & 0 & 0 \\ & X & X & X & X & 0 & 0 \\ & & X & X & X & X & 0 \\ & & & X & X & X & X \\ & 0 & & & X & X & X \\ & & & & & X & X \\ & & & & & & \delta \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{Bmatrix} = 0 \quad . \tag{31}$$

The terms in the upper right corner are zero due to bandwidth. δ is a very small number. The eigenvector may be extracted by assigning an arbitrary value (such as 1.0) to u_7 and solving successfully for u_6, u_5, etc., from the preceding rows. Note that this is equivalent to placing a load vector {F} on the right hand side that is null except for the last term which is set equal to δ.

Situations may occur in which U_{nn} is not the smallest diagonal term. Let U_{ii} be the smallest diagonal term with $i < n$. The most common reason for this occurrence is that the degrees of

ORIGINAL PAGE IS OF POOR QUALITY

freedom u_{i+1}, u_{i+2} . . . u_n are, for some reason, not coupled to the preceding degrees of freedom. In this case all of the elements in the ith row of $[U(p_j)]$ will be very small as shown below for $i = 4$, $n = 7$.

$$\begin{bmatrix} X & X & X & X & 0 & 0 & 0 \\ & X & X & X & X & 0 & 0 \\ & & X & X & X & X & 0 \\ & 0 & & \delta_{44} & \delta_{45} & \delta_{46} & \delta_{47} \\ & & & & X & X & X \\ & & & & & X & X \\ & & & & & & X \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{Bmatrix} = 0 \quad . \tag{32}$$

In the event of multiple or pathologically close eigenvalues, two or more rows or $[U(p_j)]$ will consist of very small values, exhibited below for the very exceptional case when U_{nn} is not very small.

$$\begin{bmatrix} X & X & X & X & 0 & 0 & 0 \\ & X & X & X & X & 0 & 0 \\ & & X & X & X & X & 0 \\ & 0 & & \delta_{44} & \delta_{45} & \delta_{46} & \delta_{47} \\ & & & & X & X & X \\ & & & & & \delta_{66} & \delta_{67} \\ & & & & & & X \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{Bmatrix} = 0 \quad . \tag{33}$$

In order to accommodate the exceptional cases described above with the more common case when only the last diagonal term is small, a full load vector, $\{F\}$, is used in the eigenvector calculations. A different load vector is formed for each eigenvalue to ensure that independent eigenvectors are calculated for multiple or pathologically close roots. The ith component of the vector $\{F\}$ is calculated by the following formula.

$$F_i = \frac{\delta(-1)^{ij}}{1 + \left(1 - \frac{i}{n}\right)j} \quad , \qquad (34)$$

where

δ = smallest U_{ii} ,

j = eigenvalue count ,

n = number of rows .

There is a possibility that the smallest diagonal element of [U] may be exactly zero for some eigenvalue, i.e., that an accepted eigenvalue (p_j) may be an exact eigenvalue of the problem. When this situation occurs, the zero diagonal element is set equal to a small finite number.

Eigenvectors are checked for orthogonality and normalized (see Section 9.2).

THE DETERMINANT METHOD

REFERENCE

1. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

656

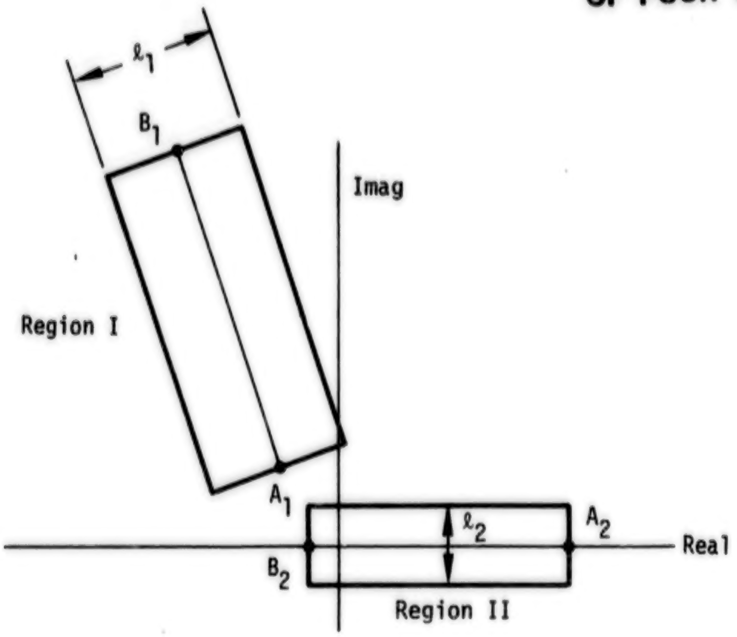

ℓ_1
B_1
Imag
Region I
A_1
A_2
ℓ_2
Real
B_2
Region II

Figure 1. Search regions.

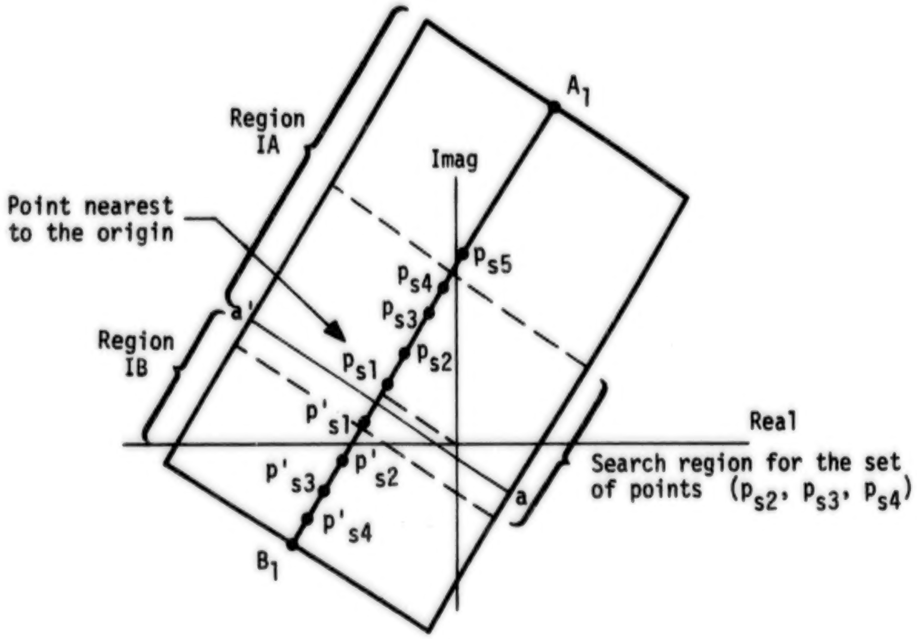

A_1
Region IA
Imag
Point nearest to the origin
p_{s5}
p_{s4}
p_{s3}
a'
Region IB
p_{s2}
p_{s1}
p'_{s1}
Real
p'_{s2}
p'_{s3}
a
Search region for the set of points (p_{s2}, p_{s3}, p_{s4})
p'_{s4}
B_1

Figure 2. Location of starting points in a region.

ORIGINAL PAGE IS
OF POOR QUALITY

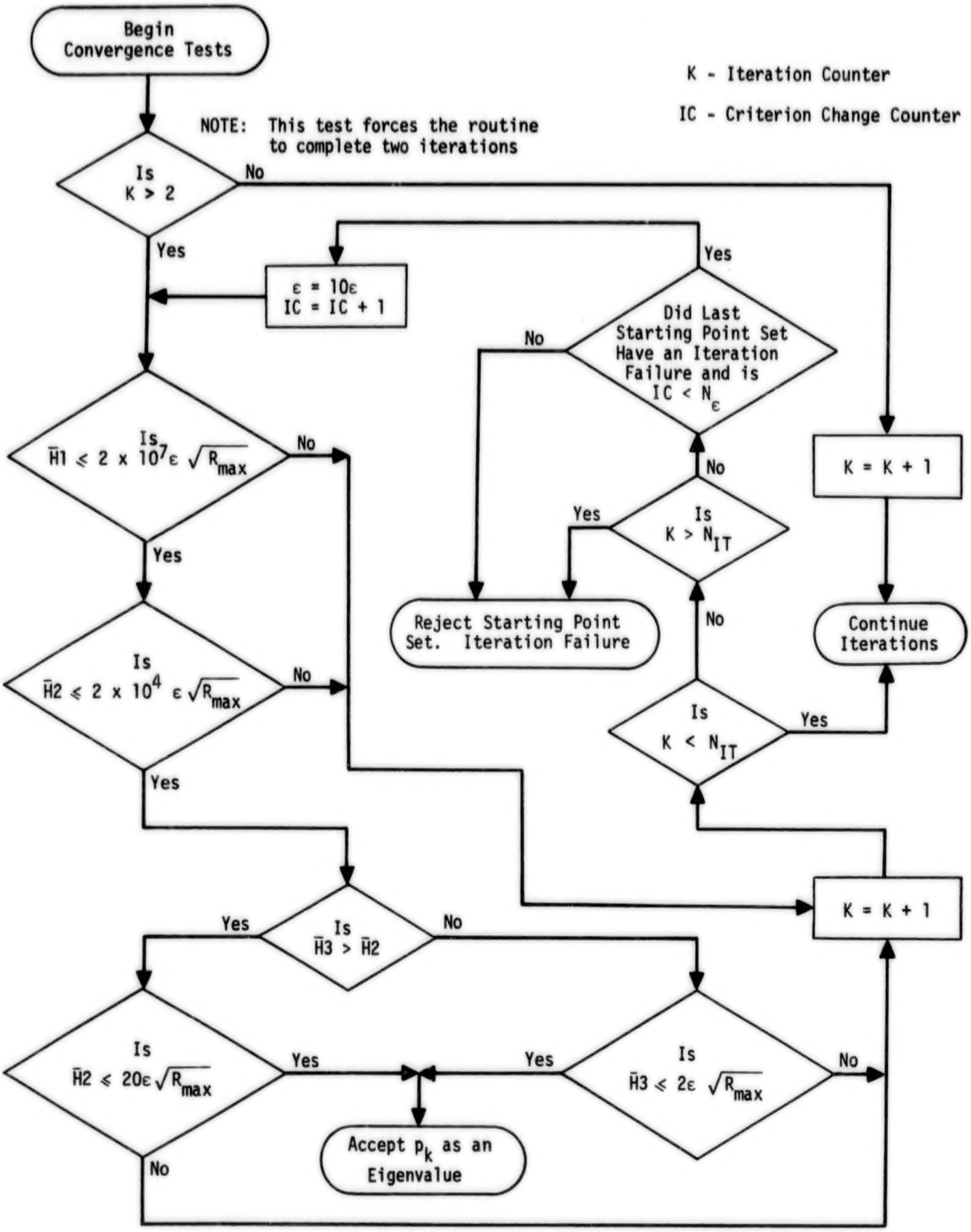

Begin Convergence Tests
K - Iteration Counter
IC - Criterion Change Counter
NOTE: This test forces the routine to complete two iterations
Is $K > 2$
No
Yes
$\varepsilon = 10\varepsilon$
$IC = IC + 1$
Yes
Did Last Starting Point Set Have an Iteration Failure and is $IC < N_\varepsilon$
No
Is $\bar{H}1 \leq 2 \times 10^7 \varepsilon \sqrt{R_{max}}$
No
Yes
$K = K + 1$
No
Yes
Is $K > N_{IT}$
Reject Starting Point Set. Iteration Failure
Continue Iterations
No
Is $\bar{H}2 \leq 2 \times 10^4 \varepsilon \sqrt{R_{max}}$
No
Yes
Is $K < N_{IT}$
Yes
$K = K + 1$
Yes
Is $\bar{H}3 > \bar{H}2$
No
Is $\bar{H}2 \leq 20\varepsilon \sqrt{R_{max}}$
Yes
Yes
Is $\bar{H}3 \leq 2\varepsilon \sqrt{R_{max}}$
No
No
Accept p_k as an Eigenvalue

Figure 3. Real eigenvalue convergence tests.

658

10.4 THE INVERSE POWER METHOD WITH SHIFTS

10.4.1 Introduction

The Inverse Power Method with Shifts is particularly effective for problems that are formulated by the displacement approach when only a fraction of all of the eigenvalues are required. The rudiments of the method are described in Wilkinson's recent book (Reference 1) where it is regarded chiefly as a powerful method for refining the accuracy of eigenvalues and eigenvectors that have been approximately located by other methods. In NASTRAN, the method is used as a stand-alone method to find all of the eigenvalues within a domain specified by the user.

It is well known that the standard inverse power method has a number of important defects for the solution of structural problems by the displacement approach. These include: awkwardness of procedure in the presence of zero eigenvalues (rigid body structural modes); slow convergence rates for closely spaced roots; and deterioration of accuracy in the higher modes as more roots are found. All of these defects are eliminated or minimized by the following modification of the method.

Let the eigenvalue problem be stated as follows,

$$[K - \lambda M]\{u\} = 0 \quad , \tag{1}$$

where, employing structural semantics, [K] is a stiffness matrix, [M] is a mass matrix, {u} is a displacement vector, and λ is the square of the natural frequency. Let

$$\lambda = \lambda_o + \Lambda \quad , \tag{2}$$

where λ_o is called the shift point. The iteration algorithm is

$$[K - \lambda_o M]\{w_n\} = [M]\{u_{n-1}\} \; , \tag{3}$$

$$\{u_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{4}$$

where C_n is equal to the value of the element of $\{w_n\}$ with largest absolute value. It is easy to prove that $1/C_n$ converges to Λ_1, the shifted eigenvalue nearest to the shift point, and that $\{u_n\}$

converges to the corresponding eigenvector of Equation 1. The following observations are pertinent.

1. A triangular decomposition of the matrix $[K - \lambda_0 M]$ is required in order to evaluate $\{w_n\}$ from Equation 3. The effort required to perform the triangular decomposition is greatly reduced if $[K - \lambda_0 M]$ is a narrow band matrix. Structural analysis by the displacement approach is characterized by narrowly banded stiffness and mass matrices.
2. It is unnecessary for $[K]$ to be nonsingular, so that rigid body modes cause no special difficulty.
3. The shift point λ_0 may be changed (at the cost of an additional triangular decomposition) in order to improve the rate of convergence toward a particular eigenvalue, or to improve accuracy and convergence rate after several roots have been extracted from a given shift point.
4. λ_0 can be placed so as to obtain the eigenvalues within a desired frequency band and not just those that have the smallest absolute value.

The Inverse Power Method with Shifts can also be applied to complex eigenvalue problems. In NASTRAN the method is applied to problems stated in the quadratic form

$$[M\lambda^2 + B\lambda + K]\{u\} = 0 \quad , \tag{5}$$

where [M], [B] and [K] may be real or complex, symmetrix or nonsymmetric, singular or nonsingular.

The development of the method is divided into separate subsections for real eigenvalue analysis, Section 10.4.2, and complex eigenvalue analysis, Section 10.4.4. In addition the procedures used by the program are summarized in Sections 10.4.3, real eigenvalue analysis, and Section 10.4.5, complex eigenvalue analysis.

10.4.2 Theory for Real Eigenvalue Analysis

10.4.2.1 Iteration Algorithm

The problem is to find the eigenvalues and eigenvectors for

$$[K - \lambda M]\{u\} = 0 \quad . \tag{1}$$

Both [K] and [M] are symmetric, but they may be singular. The eigenvalues may be either positive or negative, but we are predominantly interested in the positive eigenvalues. Eigenvalues may be multiple. It is required to find all the eigenvalues and eigenvectors within a specified range of λ.

Let

$$\lambda = \lambda_o + \Lambda \quad , \tag{2}$$

where λ_o is a constant called the shift point, so that Λ replaces λ as the eigenvalue parameter.

The algorithm used in the nth iteration step is

$$[K - \lambda_o M]\{w_n\} = [M]\{u_{n-1}\} \quad , \tag{3}$$

$$\{u_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{4}$$

where C_n is equal to the value of the element of $\{w_n\}$ with largest absolute value. It may be shown that $1/C_n$ converges to Λ_1, the shifted eigenvalue nearest to the shift point, and $\{u_n\}$ converges to the corresponding eigenvector. Convergence is proved as follows. In the proof use will be made of the orthogonality properties for the theoretically obtainable normalized exact eigenvectors, ϕ_i and ϕ_j, of Equation 1 which are

$$\phi_j{}^T M \phi_i = 0, \quad i \neq j \quad , \tag{5}$$

$$\phi_j{}^T M \phi_j = 1 \quad , \tag{6}$$

$$\phi_j{}^T K \phi_i = 0 \text{ , } i \neq j, \text{ and} \tag{7}$$

$$\phi_j{}^T K \phi_j = \lambda_j \quad . \tag{8}$$

Note that the use of brackets to indicate matrix quantities has been omitted in the interest of conciseness. Expand the trial vectors, u_n and u_{n-1}, in terms of eigenvectors:

$$u_n = \sum_i \alpha_{i,n}\phi_i \quad , \tag{9}$$

$$u_{n-1} = \sum_i \alpha_{i,n-1}\phi_i \quad , \tag{10}$$

where $\alpha_{i,n}$ and $\alpha_{i,n-1}$ are coefficients. Substitute into Equation 3, premultiply by ϕ_j^T, and invoke the orthogonality conditions. Then it is seen that

$$(\lambda_j - \lambda_o)C_n\alpha_{j,n} = \alpha_{j,n-1} \quad , \tag{11}$$

or, substituting $\lambda_j = \lambda_o + \Lambda_j$,

$$P_{j,n} = \frac{\alpha_{j,n}}{\alpha_{j,n-1}} = \frac{1}{C_n(\lambda_j - \lambda_o)} = \frac{1}{C_n\Lambda_j} \quad . \tag{12}$$

Equation 12 shows that the relative proportion, $P_{j,n}$, of an eigenvector in successive trial vectors increases in inverse proportion to the magnitude of its shifted eigenvalue. Thus the proportion of the eigenvector that is closest to the shift point increases at the expense of the others and the process converges to the eigenvector whose eigenvalue is closest to the shift point. The product $C_n\Lambda_1$ converges to unity.

10.4.2.2 Convergence Criteria

The most common method for gaging the convergence of eigenvectors is to compare the difference of the elements of successive trial vectors, each of which has been normalized so that the largest element is unity, to an arbitrarily selected small number

$$u_n - u_{n-1} < \varepsilon \quad .$$

This method is not used in NASTRAN because it is unscientific in two respects:

1. The criterion ε is not related to the mass orthogonality test which is finally used to judge the suitability of eigenvectors.
2. Normalizing on the largest element of u_n neglects the fact that the sizes of elements are largely determined by scaling, (e.g., angles are usually much smaller than translations).

Closely related methods can be devised that overcome these objections and that also predict the rate of convergence. In NASTRAN, the convergence criteria are derived from the parameter ε that is supplied by the user for the mass orthogonality test

$$\bar{\phi}_i^T M \bar{\phi}_j < \varepsilon \qquad i \neq j \quad , \tag{13}$$

where the approximate eigenvectors, $\bar{\phi}_i$, have been normalized such that

$$\bar{\phi}_i^T M \bar{\phi}_i = 1 \quad . \tag{14}$$

The mass orthogonality test is applied after all the eigenvalues have been computed.

For a pair of well-converged approximate eigenvectors that have close eigenvalues but which are well removed from all other eigenvalues:

$$\bar{\phi}_1 \simeq \phi_1 + \varepsilon_{12}\phi_2 \quad , \tag{15}$$

$$\bar{\phi}_2 \simeq \phi_2 + \varepsilon_{21}\phi_1 \quad , \tag{16}$$

since it may be assumed on the basis of Equation 12 that the power method yields negligible contributions of all other eigenvectors. Thus

$$\bar{\phi}_1^T M \bar{\phi}_2 \simeq \varepsilon_{12} + \varepsilon_{21} \quad . \tag{17}$$

The program is able to estimate ε_{12} and ε_{21} from tests that will be described. A reasonable convergence criterion to be applied after each iteration is that

$$\varepsilon_{ij} < A\varepsilon \ , \tag{18}$$

for all i and j where ε is the criterion used in the mass-orthogonality test and A is a factor, less than unity, introduced to ensure passage of the mass-orthogonality test.

It has been shown (Equation 12) that the contributions of eigenvectors to the trial vector increase from iteration to iteration in inverse proportion to their shifted eigenvalues. Thus, during the late stages of iteration for ϕ_1, it may be assumed that only two eigenvectors have significant contributions:

$$u_n = \alpha_n(\phi_1 + \sqrt{\delta_n}\,\phi_2) \quad , \tag{19}$$

where $\sqrt{\delta_n} = \varepsilon_{12}$ is introduced for later convenience and α_{1n} (see Equation 9) is shortened to α_n. By virtue of Equation 12 the relative contribution of ϕ_2 in the preceding iteration step will be larger by the ratio of the shifted eigenvalues Λ_2/Λ_1. Thus for the two previous steps

$$u_{n-1} = \alpha_{n-1}\left(\phi_1 + \frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,\phi_2\right) \ , \tag{20}$$

$$u_{n-2} = \alpha_{n-2}\left(\phi_1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,\phi_2\right). \tag{21}$$

664

ORIGINAL PAGE IS OF POOR QUALITY

It will be assumed that $|\Lambda_2|>|\Lambda_1|$.

Applying the mass orthogonality properties, Equations 5 and 6, to Equations 19 and 20, it is seen that

$$u_n^T M u_n = \alpha_n^2 (1 + \delta_n) \quad , \tag{22}$$

$$u_{n-1}^T M u_{n-1} = \alpha_{n-1}^2 \left(1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n\right) \quad . \tag{23}$$

Assuming $\left|\frac{\Lambda_2}{\Lambda_1}\right| \sqrt{\delta_n} \ll 1$ where $\left|\frac{\Lambda_2}{\Lambda_1}\right| > 1$, then to a very close approximation

$$\alpha_n \simeq \pm\left(u_n^T M u_n\right)^{1/2} \quad , \tag{24}$$

$$\alpha_{n-1} \simeq \pm\left(u_{n-1}^T M u_{n-1}\right)^{1/2} \quad . \tag{25}$$

From Equation 12, a good approximation to the eigenvalue is

$$\Lambda_1 = \frac{1}{C_n}\frac{\alpha_{n-1}}{\alpha_n} \simeq \frac{\pm 1}{C_n}\left(\frac{u_{n-1}^T M u_{n-1}}{u_n^T M u_n}\right)^{1/2} \quad , \tag{26}$$

where C_n is the normalizing factor introduced in Equation 4. The ambiguity in sign is removed by comparing the signs of selected components in u_n and u_{n-1}. A better approximation that can be evaluated when Λ_2/Λ_1 and δ_n are known is obtained by using Equations 22 and 23 for α_n and α_{n-1}.

$$\Lambda_1 = \frac{1}{C_n}\frac{\alpha_{n-1}}{\alpha_n} \simeq \pm\frac{1}{C_n}\left(\frac{u_{n-1}^T M u_{n-1}}{u_n^T M u_n}\right)^{1/2}\left(\frac{1 + \delta_n}{1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n}\right)^{1/2} \quad . \tag{27}$$

Equation 27 shows that Equation 26 is good enough because δ_n must be extremely small in order that $\varepsilon_{12} = \sqrt{\delta_n}$ pass the eigenvector convergence test. Equation 26 is used in NASTRAN.

Turning now to the evaluation of δ_n and Λ_2/Λ_1, we first define the normalized difference of successive trial vectors,

$$\delta u_n = \frac{u_n}{\left(u_n^T M u_n\right)^{1/2}} - \frac{u_{n-1}}{\left(u_{n-1}^T M u_{n-1}\right)^{1/2}} \simeq \frac{u_n}{\alpha_n} - \frac{u_{n-1}}{\alpha_{n-1}} \quad . \tag{28}$$

Then, from Equations 19, 20 and 21:

$$\delta u_n = \left(1 - \frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\,\phi_2 \quad , \tag{29}$$

$$\delta u_{n-1} = \frac{\Lambda_2}{\Lambda_1}\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\;\phi_2 \quad , \tag{30}$$

and, by virtue of Equation 6

$$(\delta u_n)^T M \delta u_n = \left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n \quad , \tag{31}$$

$$(\delta u_n)^T M \delta u_{n-1} = \frac{\Lambda_2}{\Lambda_1}\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n \quad , \tag{32}$$

so that

$$\frac{\Lambda_2}{\Lambda_1} = \frac{(\delta u_n)^T M \delta u_{n-1}}{(\delta u_n)^T M \delta u_n} \quad , \tag{33}$$

and

$$\delta_n = \frac{(\delta u_n)^T M \delta u_n}{\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2} \quad . \tag{34}$$

The convergence criterion is

$$\sqrt{\delta_n} < A\varepsilon \quad . \tag{35}$$

Note that the application of the criterion requires three successive trial vectors. In some cases the rate of convergence may be so rapid that Λ_2/Λ_1, as computed by Equation 33, may have lost all numerical significance. In order to avoid this difficulty and also to improve efficiency in the event of rapid convergence, a test based on only two trial vectors is applied first.

The rapid convergence test is based on the fact that the sweeping procedures, Section 10.4.2.4, guarantee the orthogonality of very close eigenvalues. It is only necessary to avoid contamination with eigenvectors that are not in the cluster. Thus if

$$\left|\frac{\lambda_2 - \lambda_1}{\lambda_1}\right| = \left|\frac{\Lambda_2 - \Lambda_1}{\Lambda_1 + \lambda_o}\right| < \gamma \quad , \tag{36}$$

where γ is a small parameter built into the program, then it is not required that Equation 35 be satisfied.

If the criterion obtained by multiplying Equations 35 and 36 together, namely

$$\left|\frac{\Lambda_2 - \Lambda_1}{\Lambda_1 + \lambda_o}\right| \sqrt{\delta_n} < A\varepsilon\gamma \quad , \tag{37}$$

is satisfied, then either Equation 35 or 36, or both, are satisfied. From Equation 34

$$\left|\frac{\Lambda_2 - \Lambda_1}{\Lambda_1 + \lambda_o}\right| \sqrt{\delta_n} = \frac{\left[(\delta u_n)^T M (\delta u_n)\right]^{1/2}}{\left|1 + \frac{\lambda_o}{\Lambda_1}\right|} \quad . \tag{38}$$

The rapid convergence test is, therefore

$$\eta = \left[(\delta u_n)^T M \delta u_n\right]^{1/2} < A\varepsilon\gamma \left|1 + \frac{\lambda_c}{\Lambda_1}\right| \quad . \tag{39}$$

Λ_1 is evaluated from Equation 26. The test requires only two successive trial vectors, and it is performed as often as possible. The value of γ stored in the program is .01.

10.4.2.3 Change of Shift Point

The rate of convergence of the iteration algorithm can be estimated from the parameters, Λ_1, Λ_2, and δ_n. From Equations 19 and 20, the estimated proportion of the second eigenvector in the (n+k)th trial eigenvector is

$$\sqrt{\delta_{n+k}} = \left(\frac{\Lambda_1}{\Lambda_2}\right)^k \sqrt{\delta_n} \quad . \tag{40}$$

If the (n+k)th trial eigenvector satisfies the convergence criterion, Equation 35, then

$$\left(\frac{\Lambda_1}{\Lambda_2}\right)^k < \frac{A\varepsilon}{\sqrt{\delta_n}} \quad , \tag{41}$$

and the estimate of the required number of additional iterations, beyond the nth, is

$$k > \frac{\log\left(\frac{\sqrt{\delta_n}}{A\varepsilon}\right)}{\log\left|\frac{\Lambda_2}{\Lambda_1}\right|} > k - 1 \quad . \tag{42}$$

If $|\Lambda_2/\Lambda_1|$ is nearly equal to unity, the required number of iterations may be very large.

Suppose, for example, that

$$\varepsilon = 10^{-6}, \quad A = 10^{-1}, \quad \delta_n = 10^{-4}, \quad \frac{\Lambda_2}{\Lambda_1} = 1.10 \ . \quad \text{Then}$$

$$k > \frac{\log(10^5)}{\log(1.1)} = 119 \quad .$$

If, however, the shift point is changed to the current estimate of the eigenvalue, $\Lambda_{1,n} + \lambda_o$, the number of additional iterations will usually be quite small. The error in the calculation of Λ_1 from Equation 26 is of the order of δ_n, as may be seen by comparing Equation 26 and Equation 27. Thus assuming the new shift point to be located at $\Lambda_1(1-\delta_n)$, the ratio of the first and second eigenvalues from the new shift point is

$$\frac{\Lambda_2'}{\Lambda_1'} = \frac{\Lambda_2 - \Lambda_1(1-\delta_n)}{\Lambda_1 - \Lambda_1(1-\delta_n)} = \frac{\Lambda_2 - \Lambda_1}{\delta_n \Lambda_1} \quad . \tag{43}$$

Using this result in the previous example, the estimated number of iterations from the new shift point is

$$k' = \frac{\log(10^5)}{\log(10^3)} = 1.6 \quad .$$

In many cases the time required to complete the estimated number of iterations from the old shift point may exceed the time required to make a triangular decomposition and to complete the iteration from the new shift point. In such cases a decision to shift is made at the earliest time that it can reliably be made.

Let T_d be the time for one triangular decomposition and let T_i be the time for one iteration. Then, if

$$T_d < (k - k')T_i \quad , \tag{44}$$

a new shift point is selected. The minimum estimate of k' is 2, to permit testing for convergence after the shift. The relative times for decomposition and iteration depend on the bandwidth of the dynamic matrix. As a first approximation

$$\frac{T_d}{T_i} \simeq \frac{B}{3} \quad , \tag{45}$$

where B is the semibandwidth. Thus, as a first approximation, a new shift point should be selected if

$$B < 3(k - k') . \tag{46}$$

In practice accurate values of T_d and T_i are available to the computer because at least one triangular decomposition and several iterations have already been performed.

Although the test described above will predict the rate of convergence, it is not foolproof for two reasons. The first is the requirement that the proportion in the trial vectors of all eigenvectors except the two lowest be very small. The second is that there must be a sufficient proportion of the second eigenvector to permit reasonably reliable calculation of Λ_2. A check on the convergence criteria is provided by examining the series of values of Λ_1 and Λ_2. If the differences between successive values decrease monotonically, we are assured of uniform convergence. On the other hand, a sudden jump in the value of Λ_2 will indicate that the limit of digital precision has been reached.

Two additional tests are performed in NASTRAN to improve reliability.

Define the normalized increments in the approximate eigenvalues from successive iterations

$$\bar{h}_{1,n} = \frac{\lambda_{1,n} - \lambda_{1,n-1}}{R_0} = \frac{1}{R_0}(\Lambda_{1,n} - \Lambda_{1,n-1}) \quad , \tag{47}$$

and

$$\bar{h}_{2,n} = \frac{\lambda_{2,n} - \lambda_{2,n-1}}{\lambda_0} = \frac{(\Lambda_{2,n} - \Lambda_{2,n-1})}{\lambda_0} \quad , \tag{48}$$

where R_0 is a normalizing factor to be specified later (see Equation 69). The shift test, Equation 44, is applied provided that

$$|\bar{h}_{1,n}| < \varepsilon_1 \quad . \tag{49}$$

The criterion ε_1 is not made very severe. The reason for the criterion is that, if we decide to shift, we wish some assurance that the new shift point is substantially better than the original shift point.

The second test relates to the reliability of Λ_2. If

$$\varepsilon_2 > |\bar{h}_{2,n}| > |\bar{h}_{2,n-1}| \quad , \tag{50}$$

then $\Lambda_{2,n-1}$ will be used in all subsequent calculations of δ from Equation 34. Otherwise continue to compute Λ_2 at each step. Note that the ratio Λ_2/Λ_1 from Equation 33 is no longer required after the ε_2 test has failed.

10.4.2.4 Sweeping of Previously Found Eigenvectors

The trial vectors must be swept to eliminate contributions due to previously found eigenvalues that are closer to the shift point than the current eigenvalue. The algorithm is developed as follows. Let the unswept trial vector be $\bar{u}_n$ which, from Equation 9, can be represented as:

$$\bar{u}_n = \sum_{i=1}^{m} \alpha_{i,n}\phi_i + u_n \quad , \tag{51}$$

where ϕ_i, i = 1,2...m, are exact values of the previously extracted eigenvalues and u_n is the swept trial vector which will include only contributions from the remaining modes. The computed values for previously extracted eigenvectors are

$$\bar{\phi}_i = \frac{u_{i,N}}{\sqrt{u_{i,N}^T M u_{i,N}}} \quad , \tag{52}$$

where $u_{i,N}$ is the last (N^{th}) trial vector obtained in the iteration for λ_i.

If $\bar{\phi}_i$ were an exact eigenvector, then, from the mass-orthogonality properties,

$$\alpha_{i,n} = \bar{\phi}_i^T M \bar{u}_n \quad , \tag{53}$$

and

$$u_n = \bar{u}_n - \sum_i \left(\bar{\phi}_i^T M \bar{u}_n \right) \bar{\phi}_i \quad , \tag{54}$$

which is the sweeping algorithm.

Let us examine the error in the process. Suppose

$$\bar{\phi}_i = \phi_i + \varepsilon\phi_j \quad , \tag{55}$$

where ϕ_i and ϕ_j are exact eigenvectors and $\bar{\phi}_i$ is the eigenvector that is currently being extracted. Also let

$$\bar{u}_n = \alpha_{i,n}\phi_i + \alpha_{j,n}\phi_j \quad . \tag{56}$$

Thus,

$$(\bar{\phi}_i^T M \bar{u}_n) = \alpha_{i,n} + \epsilon\alpha_{j,n} \quad , \tag{57}$$

and, considering only one previously swept eigenvector in Equation 54

$$\begin{aligned} u_n &= \alpha_{i,n}\phi_i + \alpha_{j,n}\phi_j - (\alpha_{i,n} + \epsilon\alpha_{j,n})(\phi_i + \epsilon\phi_j) \\ &= \left(\alpha_{j,n}(1 - \epsilon^2) - \epsilon\alpha_{i,n}\right)\phi_j - \epsilon\alpha_{j,n}\phi_i \quad . \end{aligned} \tag{58}$$

Redefine the coefficients of ϕ_i and ϕ_j in Equation 58

$$u_n = \bar{\alpha}_{j,n}\phi_j + \bar{\alpha}_{i,n}\phi_i \quad . \tag{59}$$

The ratio of the relative amplitudes of eigenvector components in the unswept vectors obtained after the next iteration is, for $\epsilon \ll 1$,

$$\frac{\alpha_{i,n+1}}{\alpha_{j,n+1}} = \frac{\Lambda_j}{\Lambda_i}\frac{\bar{\alpha}_{i,n}}{\bar{\alpha}_{j,n}} \simeq -\epsilon\frac{\Lambda_j}{\Lambda_i} \quad . \tag{60}$$

Convergence to ϕ_j rather than to ϕ_i requires that

$$\left|\epsilon\frac{\Lambda_j}{\Lambda_i}\right| < 1 \quad . \tag{61}$$

If all previous eigenvectors are reasonably accurate, we need only to be concerned for cases in which $\Lambda_j \gg \Lambda_i$. This is, however, precisely the condition that exists at the shifted shift point, λ_o'. Thus, λ_o' should not be used for obtaining additional eigenvalues.

Furthermore, if an eigenvalue is found to be very close to the original shift point, then the shift point should be changed in the search for other eigenvalues. The criterion for changing starting points (original shift points) may be written

$$\epsilon\left|\frac{R_o}{\lambda_o' - \lambda_1}\right| > \epsilon_3 \quad , \tag{62}$$

where ϵ is the coefficient specified in the mass-orthogonality test, Equation 13, λ_1 is the first eigenvalue found from the starting point λ_o, R_o is the distance from λ_o to the farthest eigenvalue to be extracted from λ_o, and ϵ_3 is a safety factor. Suppose, for example, $\epsilon = 10^{-6}$ and $\epsilon_3 = 10^{-2}$. Then the starting point should be changed if

$$\left|\frac{\lambda_o - \lambda_1}{R_o}\right| < 10^{-4} \quad . \tag{63}$$

Another conclusion to be derived from Equation 62 is that the trial vectors should be swept at each iteration. If, for example, trial vectors were swept every k iterations, proper convergence would require that

$$\varepsilon\left|\frac{R_o}{\lambda_o - \lambda_1}\right|^k < 1 \quad , \tag{64}$$

which, if k were larger than about 3 or 4, would require continuous changing of starting points.

10.4.2.5 Initial Trial Vectors and the Extraction of Multiple Eigenvectors

The choice of a trial vector for finding the first eigenvalue from any starting point is clearly arbitrary. The vector used in NASTRAN has a broad spectrum with regard to spatial distribution in order to provide efficient extraction of eigenvalues in the complete wavelength range.

There is an advantage in finding successive eigenvalues from the same starting point, in using one of the trial vectors employed in a late stage of the previous iteration. The reason for the advantage is that such a vector, after having been swept, is quite "rich" in the eigenvector that is next closest to the starting point and will, therefore, converge more rapidly. There is a difficulty, however, in that, if the previous eigenvalue is a multiple eigenvalue, a return to it will probably be prevented since the swept trial vector contains no component of the multiple vector(s). If the process of transforming trial vectors is repeated often enough, however, the process will eventually return to the multiple eigenvalue by virtue of the growth of round-off error. Unfortunately this cannot be relied upon. The safer course, which is to use a new initial trial vector for each eigenvalue, will require more iterations since there is no "enrichment".

The following scheme, in which we might have our cake and eat it too, is used in NASTRAN:

1. Find the first eigenvalue from a starting point with an arbitrarily selected trial vector.
2. Use the next to last trial vector obtained at the starting point during the first iteration sequence as the initial trial vector for the second iteration sequence. This will ensure a very large proportion of the next closest eigenvector. Since we are

required to save two vectors at every iteration step in order to check convergence, this should be no burden on storage requirements. Furthermore, since all calculations are done in double precision, and only single-precision convergence of eigenvectors can be required, the swept vector should not contain much random noise.

3. Continue until an eigenvalue is found that is outside the prescribed range of the starting point (see Section 10.4.2.6).
4. At this stage, use a new arbitrarily selected initial vector and find one eigenvalue.
5. If the eigenvalue found in step 4 is outside the range of the starting point, go to a new starting point. If it is not outside the range, repeat steps 2, 3 and 4 until the eigenvalue found in step 4 is outside the range. The maximum number of times that steps 2, 3 and 4 will be performed is equal to the multiplicity of the eigenvalue with greatest multiplicity.

An exception to the above sequence occurs if step 1 yields an eigenvalue outside the range. In that case, the program goes directly to a new starting point.

10.4.2.6 Distribution of Starting Points

For Vibration Analysis the user specifies:

1. The maximum frequency of interest, f_{max}.
2. The minimum frequency of interest, f_{min}.
3. The estimated number of modes in the range, N_e.
4. The desired number of modes, N_d, to be computed.

For Buckling Analysis the user specifies:

1. The maximum eigenvalue of interest, λ_{max}.
2. The minimum eigenvalue of interest, λ_{min}, which may be negative.
3. The estimated number of modes in the range, N_e.
4. The desired number of modes, N_d, to be computed.

For Vibration Analysis, the eigenvalue

$$\lambda = (2\pi f)^2 = \omega^2 \quad . \tag{65}$$

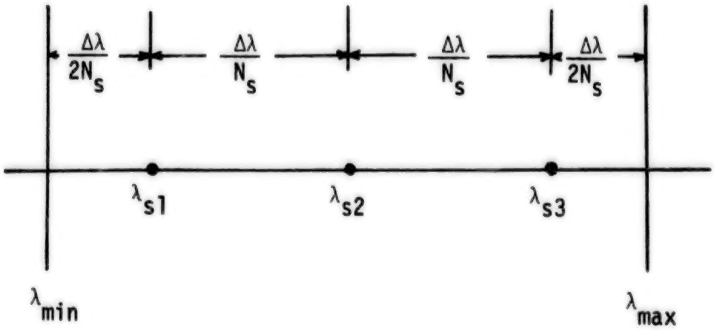
If the desired number of modes, N_d, is not specified, then all modes in the range will be found provided that there are not more than $3N_e$ modes. If the desired number of eigenvalues is less than the expected number, those that are closest to zero will be computed.

Starting points are distributed on the assumption that the eigenvalues are uniformly distributed with respect to λ, (frequency-squared), and that about six eigenvalues will be found from each starting point.

The number of starting points, N_s, is selected such that

$$N_s - 1 < \frac{N_e}{6} < N_s \quad , \tag{66}$$

and they are distributed as follows.


$\frac{\Delta\lambda}{2N_s}$
$\frac{\Delta\lambda}{N_s}$
$\frac{\Delta\lambda}{N_s}$
$\frac{\Delta\lambda}{2N_s}$
λ_{s1}
λ_{s2}
λ_{s3}
λ_{min}
λ_{max}

where $\Delta\lambda = \lambda_{max} - \lambda_{min} = (2\pi f_{max})^2 - (2\pi f_{min})^2$. Thus the nth starting point,

$$\lambda_{sn} = \lambda_{min} + \frac{n - \frac{1}{2}}{N_s} \Delta\lambda \quad . \tag{67}$$

The search-ranges for the starting points are selected to overlap each other by 10%. Thus, the search-range for shifted eigenvalues, $\Lambda = \lambda - \lambda_{sn}$, is

$$-R_o < \Lambda < +R_o \quad , \tag{68}$$

where

$$R_o = .55 \frac{\Delta\lambda}{N_s} \quad . \qquad (69)$$

R_o is also used both in deciding whether or not to relocate starting points, Equation 62, and in the ε_1 preliminary convergence test, Equation 47.

10.4.2.7 Termination

The process of finding eigenvalues can be terminated for any of the following reasons:

1. All of the eigenvalues that the problem contains have been found.
2. All of the eigenvalues in the desired range, $\lambda_{min} < \lambda < \lambda_{max}$, have been found.
3. The desired number of eigenvalues, N_d, have been found.
4. The number of eigenvalues that have been found is larger by some factor than the number estimated to be in the desired range.

Reason 1 is imperative because, if the iteration process is continued, it will converge on eigenvalues that it has already found and give false results. A reasonable, although not foolproof, test for whether or not all (finite) eigenvalues have been found is to compute the total number of eigenvalues from

$$N_t = N_{eq} - N_o - N_{oM} \quad , \qquad (70)$$

where

N_{eq} = number of rows in the dynamic matrix.

N_o = number of zero eigenvalues prescribed by the user, and found by separate procedures.

N_{oM} = number of diagonal terms of the mass matrix that are equal to zero.

The test is not foolproof because the mass matrix can be singular even if none of the diagonal terms are zero. The detection of general singularity is regarded as too difficult to be worthwhile for the present application.

Reason 2 is detected quite simply by applying the range test to the last starting point.

The recommended value for the factor mentioned in reason 4 is 3.0.

Once the process of extracting eigenvalues has terminated, all of the eigenvalues and eigenvectors that have been found (and not just the desired number or those within the desired range) are prepared for output. If eigenvalues are to be used in a modal solution they will all be positive and only the desired number closest to the origin (if that number has been specified), or the total number within the specified frequency range, whichever is less, are used.

10.4.3 Summary of Procedures for Real Eigenvalue Analysis

Flow diagrams illustrating recommended procedures are shown in Figures 1, 2 and 3. The procedures involved in each block are summarized below.

1. Compute Distribution of Starting Points (see Section 10.4.2.6)

a. Select number of starting points, N_s, by:

$$N_s - 1 < \frac{N_e}{6} < N_s \quad , \tag{1}$$

where N_e is the estimated number of eigenvalues in the frequency range of interest.

b. Locate starting points by:

$$\lambda_{sn} = \lambda_{min} + \frac{n - \frac{1}{2}}{N_s} \Delta\lambda \quad , \tag{2}$$

where

$$\Delta\lambda = \lambda_{max} - \lambda_{min} = (2\pi f_{max})^2 - (2\pi f_{min})^2 \quad . \tag{3}$$

2. Select a Starting Point

The first starting point is λ_{s1}, the second starting point is λ_{s2}, etc., until all starting points are used up or until the desired number, N_d, of roots has been found.

3. Select an Arbitrary Starting Vector

Each arbitrary starting vector is distinct from all preceding starting vectors. The vector should have a broad spectrum with regard to spatial distribution (i.e., not concentrated in the long wavelength or in the short wavelength range).

4. Compute One Eigenvalue and One Eigenvector

See flow diagram for this block, Figure 2.

4.1 Decompose Dynamic Matrix

The matrix to be decomposed is $[K - \lambda_o M]$ which is real and symmetric and where λ_o is either a starting point, λ_s, a moved starting point, λ_s', or a shifted shift point, λ_o'. Real arithmetic without pivoting is used. Partial pivoting is available on an optional basis.

ORIGINAL PAGE IS OF POOR QUALITY

4.2 One Vector Iteration (see Sections 10.4.2.1 and 10.4.2.4)

The iteration algorithm is:

$$[K - \lambda_o M]\{w_n\} = [M]\{u_{n-1}\} \equiv \{F_{n-1}\} \quad , \tag{4}$$

$$\{\bar{u}_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{5}$$

$$\{u_n\} = \{\bar{u}_n\} - \sum_i [\phi_i{}^T M \bar{u}_n]\{\phi_i\} \; , \tag{6}$$

where C_n is equal to the element of $\{w_n\}$ with largest magnitude and $\{\phi_i\}$ is a normalized eigenvector, previously found:

$$\{\phi_i\} = \frac{\{u_{i,N}\}}{\sqrt{u_{i,N}{}^T M u_{i,N}}} \quad . \tag{7}$$

$\{u_{i,N}\}$ is the last vector found in iterating for the ith eigenvalue. The sum on (i) in Equation 6 extends over all eigenvectors previously found, including rigid body modes found in separate procedures. The sweeping operation, Equation 6, is also applied to starting vectors, $\{\bar{u}_o\}$.

4.3 Convergence Tests (see Section 10.4.2.2)

See flow diagram for this block, Figure 3.

a. The quantities Λ_1, Λ_2, η, and δ are computed as follows.

1. Form
$$\{F_n\} = [M]\{u_n\} \quad , \tag{8}$$
which is also required in the next iteration (see Equation 4).

2. Compute
$$\alpha_n = (\{u_n\}^T\{F_n\})^{1/2} \quad . \tag{9}$$

3. Compute
$$\{\delta u_n\} = \frac{\{u_n\}}{\alpha_n} - \frac{\{u_{n-1}\}}{\alpha_{n-1}} \quad . \tag{10}$$

4. Compute
$$\{\delta F_n\} = \frac{\{F_n\}}{\alpha_n} - \frac{\{F_{n-1}\}}{\alpha_{n-1}} \quad . \tag{11}$$

Then

5.
$$\Lambda_1 = \frac{1}{C_n}\frac{\alpha_{n-1}}{\alpha_n} \quad . \tag{12}$$

678

6.

$$\eta = (\{\delta u_n\}^T\{\delta F_n\})^{1/2} \quad , \qquad , \qquad (13)$$

used in the rapid convergence test.

7.

$$\frac{\Lambda_2}{\Lambda_1} = \frac{\{\delta u_n\}^T\{\delta F_{n-1}\}}{\{\delta u_n\}^T\{\delta F_n\}} \quad . \qquad (14)$$

8.

$$\delta = \frac{\{\delta u_n\}^T\{\delta F_n\}}{\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2} \quad . \qquad (15)$$

except that the maximum value of $\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2$ is limited to 10.0.

b. In the rapid convergence test,

$$\eta < A\varepsilon\gamma \left| 1 + \frac{\lambda_o}{\Lambda_1} \right| \quad , \qquad (16)$$

ε = criterion specified by the user for the mass-orthogonality check,

γ = close root criterion (= 10^{-2}),

A = safety factor (= 10^{-1}).

The default value of ε is 10^{-4}.

c. The test sequence

$$\eta_{n-1} < 10^{-6} \quad , \qquad (17)$$

$$\eta_n > 1.01\, \eta_{n-1} \quad , \qquad (18)$$

is introduced to provide for situations in which round-off error dominates the convergence criteria. These tests were not discussed in Section 10.4.2.

d. The ε_2 test is:

If $\varepsilon_2 > |\bar{h}_{2,n}| > |\bar{h}_{2,n-1}|$: Fail . (19)

Otherwise: Pass.

where, using successive values of Λ_2 computed by Equation 14,

$$\bar{h}_{2,n} = \frac{\Lambda_{2,n} - \Lambda_{2,n-1}}{\lambda_o} \quad . \qquad (20)$$

The value of ε_2 stored in the program is .02.

If the ε_2 test fails, the value of Λ_2 is set equal to $\Lambda_{2,n-1}$ in calculating δ (Equation 15) and is not changed until the shift point is changed or an eigenvalue is found.

4.4 ε_1 Test (see Section 10.4.2.3)

The ε_1 test is

$$\text{if} \qquad \left| \frac{\Lambda_{1,n} - \Lambda_{1,n-1}}{R_o} \right| < \varepsilon_1 \ : \ \text{Pass . Proceed to 4.5.} \qquad (21)$$

Otherwise: Fail. Return to 4.2.

Where

$$R_o = .55 \frac{\Delta\lambda}{N_s} \quad . \qquad (22)$$

The value of ε_1 stored in the program is 10^{-3}. The ε_1 test is also applied as part of the rapid convergence procedure, see Figure 2.

4.5 Shift Decision (see Section 10.4.2.3)

The criterion for shifting is

$$\text{if} \qquad T_d < (k - 2)T_i \ , \qquad (23)$$

or if for two successive iterations

$$\left| \frac{\Lambda_2}{\Lambda_1} \right| \leq 1 \ : \ \text{Shift.}$$

Otherwise: Continue to iterate.

680

Where

T_d = estimated time for one triangular decomposition,

T_i = estimated time for one vector iteration,

k = estimated number of additional iterations.

k is evaluated from

$$k = \frac{\ln \sqrt{\frac{|\delta|}{A\varepsilon}}}{\ln \left| \frac{\Lambda_2}{\Lambda_1} \right|} \quad . \qquad (24)$$

4.6 Change Shift Point

Change shift point to

$$\lambda_o' = \lambda_o + \Lambda_1 \quad , \qquad (25)$$

where Λ_1 is computed by Equation 12.

4.7 Reset Iteration Counter

Set $n = 0$, and also remove the flag that the ε_2 test has failed.

5. Are We Done?

The answer is yes if:

a. The total desired number of eigenvalues, N_d, have been found, or

b. The number found exceeds three times the estimated number, N_e, or

c. The number of eigenvalues found equals the total number in the problem. The total number in the problem is computed from

$$N_t = N_{eq} -- N_o - N_{oM} \quad , \qquad (26)$$

where

N_{eq} = number of rows in the dynamic matrix,

N_o = number of zero eigenvalues prescribed by the user and found by separate procedures,

N_{oM} = number of diagonal terms of the mass matrix that are equal to zero.

6. Is Eigenvalue Outside Range? (see Section 10.4.2.6)

The test is:

$$-R_o < \lambda_i - \lambda_s < +R_o \quad : \quad \text{Inside range.} \tag{27}$$

Otherwise: Outside range.

7. Select Next-to-Last Trial Vector as Starting Vector (see Section 10.4.2.5)

   a. If there was no shift in finding the previous eigenvalue (i.e., if Equation 23 has never been satisfied) set:

$$\bar{u}_{i+1,o} = u_{i,N-1} \quad . \tag{28}$$

   b. If there were one or more shifts, set $\bar{u}_{i+1,o}$ equal to last trial vector before the first shift.

8. Too Close to Starting Point? (see Section 10.4.2.4)

The criterion is:

$$\varepsilon \left| \frac{R_o}{\lambda_i - \lambda_s} \right| > \varepsilon_3 \quad : \quad \text{Too close} \quad , \tag{29}$$

where

ε = criterion used in mass orthogonality test,

λ_i = eigenvalue just found,

λ_s = starting point,

R_o = range, see Equation 22.

The value for ε_3 stored in the program is .05.

9. Move Starting Point

Change starting point location to

$$\lambda_s' = \lambda_s \pm .02\, R_o \quad , \tag{30}$$

where the (+) or (-) is selected to move the starting point away from the eigenvalue.

10. Is This the First Eigenvalue Found Since the Arbitrary Starting Point Vector Was Changed?

See Section 10.4.2.5 on Initial Trial Vectors.

11. Last Starting Point?

If the answer is yes, we are done. If the answer is no, select a new starting point.

10.4.4 Theory for Complex Eigenvalue Analysis

10.4.4.1 Iteration Algorithm

The problem is to find the eigenvalues and eigenvectors for

$$[Mp^2 + Bp + K]\{u\} = 0 \quad , \tag{1}$$

where $\boxed{M}$, $\boxed{B}$, and $\boxed{K}$ may be real or complex, symmetric or nonsymmetric, singular or nonsingular. Eigenvalues may be multiple. All the eigenvalues within specified regions of the p-plane are to be found.

Define the velocity vector

$$\{v\} = p\{u\} \quad . \tag{2}$$

Equation 1 may then be rewritten in partitioned form as

$$p \left[\begin{array}{c|c} M & 0 \\ \hline 0 & I \end{array}\right] \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} + \left[\begin{array}{c|c} B & K \\ \hline -I & 0 \end{array}\right] \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} = 0 \quad . \tag{3}$$

Let

$$p = \lambda_o + \Lambda \quad , \tag{4}$$

where the constant λ_o is called the shift point. Substituting Equation 4 into Equation 3:

$$\left[\begin{array}{c|c} -B - M\lambda_o & -K \\ \hline I & -\lambda_o I \end{array}\right] \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} = \Lambda \left[\begin{array}{c|c} M & 0 \\ \hline 0 & I \end{array}\right] \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} \quad , \tag{5}$$

or, assuming for the moment that $\boxed{M}$ is nonsingular:

$$\left[\begin{array}{c|c} -M^{-1}B - \lambda_o I & -M^{-1}K \\ \hline I & -\lambda_o I \end{array}\right] \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} = \Lambda \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} \quad . \tag{6}$$

PAGE 10.4-23B INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

685

Equation 6 is an eigenvalue problem in standard form, i.e.,

$$[A]\{x\} = \Lambda\{x\} \quad , \tag{7}$$

where

$$\{x\} = \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} \quad .$$

The standard iteration algorithm for the inverse power method is

$$[A]\{x_n\} = \frac{1}{C_n}\{x_{n-1}\} \quad , \tag{8}$$

where C_n is a normalizing factor to be determined. Convergence of the algorithm is discussed in Section 10.4.4.3.

An alternate form of the iteration algorithm corresponding to Equation 5 is:

$$\left[\begin{array}{c:c} -B - \lambda_o M & -K \\ \hdashline I & -\lambda_o I \end{array}\right] \left\{\begin{array}{c} v_n \\ \hdashline u_n \end{array}\right\} = \frac{1}{C_n} \left[\begin{array}{c:c} M & 0 \\ \hdashline 0 & I \end{array}\right] \left\{\begin{array}{c} v_{n-1} \\ \hdashline u_{n-1} \end{array}\right\} \quad . \tag{9}$$

A form that is more convenient for computation is obtained as follows: From the second row of Equation 9, suppressing matrix brackets for conciseness,

$$v_n = \lambda_o u_n + \frac{1}{C_n} u_{n-1} \quad . \tag{10}$$

Substitute Equation 10 into the top row of Equation 9:

$$-(B + \lambda_o M)(\lambda_o u_n + \frac{1}{C_n} u_{n-1}) - K u_n = \frac{1}{C_n} M v_{n-1} \quad , \tag{11}$$

or

$$(\lambda_o{}^2 M + \lambda_o B + K) u_n = -\frac{1}{C_n}\Big((B + \lambda_o M) u_{n-1} + M v_{n-1}\Big) \quad . \tag{12}$$

Let

$$w_n = C_n u_n \quad . \tag{13}$$

The complete algorithm is, then

$$(\lambda_o^2 M + \lambda_o B + K)w_n = -(B + \lambda_o M)u_{n-1} - Mv_{n-1} ,$$

$$u_n = \frac{1}{C_n} w_n , \tag{14}$$

$$v_n = \lambda_o u_n + \frac{1}{C_n} u_{n-1} .$$

C_n is selected equal to the element of w_n with largest magnitude. Note that triangular decomposition of the dynamic matrix $[p^2M + pB + K]$ is required at the point $p = \lambda_o$.

10.4.4.2 Orthogonality Properties for Nonsymmetric Matrices

Development of orthogonality properties is required in order to prove convergence and to derive sweeping techniques. Consider the pair of eigenvalue problems

$$Ax = \Lambda x , \tag{15}$$

$$A^T y = \Lambda y , \tag{16}$$

where A is a general, nonsymmetric, real or complex matrix. The eigenvalues of Equation 15 are the zeroes of the determinant $|A - \Lambda I|$. The eigenvalues of Equations 15 and 16 are the same because

$$|A - \Lambda I| = |(A^T - \Lambda I)^T| = |(A^T - \Lambda I)| . \tag{17}$$

The eigenvectors of Equations 15 and 16 are not, however, identical. The eigenvectors satisfy

$$(A - \Lambda_i)\phi_i = 0 , \tag{18}$$

$$(A^T - \Lambda_j)\overline{\overline{\phi}}_j = 0 . \tag{19}$$

ϕ_i is called a right eigenvector of A and $\overline{\overline{\phi}}_j$ is called a left eigenvector of A because, transposing Equation 19,

689

$$\bar{\bar{\phi}}_j^T(A - \Lambda_j) = 0 \qquad (20)$$

The left and right eigenvectors are orthogonal for $\Lambda_i \neq \Lambda_j$. To show this, premultiply Equation 18 by $\bar{\bar{\phi}}_j^T$

$$\bar{\bar{\phi}}_j^T A\phi_i - \Lambda_i\bar{\bar{\phi}}_j^T\phi_i = 0 \quad , \qquad (21)$$

or, using Equation 20

$$(\Lambda_j - \Lambda_i)\bar{\bar{\phi}}_j^T\phi_i = 0 \quad , \qquad (22)$$

so that, for $\Lambda_i \neq \Lambda_j$

$$\bar{\bar{\phi}}_j^T\phi_i = 0 \quad , \qquad (23)$$

which is known as the property of biorthogonality.

Equation 23 will be useful in proving convergence of the iteration algorithm. It is not useful for sweeping procedures because transposition of the A matrix in Equation 6 leads to a form that is extremely awkward for computation. A more convenient form of orthogonality property is developed as follows. Let ϕ_i be an eigenvector satisfying

$$(p_i^2M + p_iB + K)\phi_i = 0 \quad , \qquad (24)$$

and $\bar{\bar{\phi}}_j$ be an eigenvector satisfying

$$\bar{\bar{\phi}}_j^T(p_j^2M + p_jB + K) = 0 \quad . \qquad (25)$$

Premultiply Equation 24 by $p_j\bar{\bar{\phi}}_j^T$, postmultiply Equation 25 by $p_i\phi_i$, and subtract;

$$p_j\bar{\bar{\phi}}_j^T(p_i^2M + p_iB + K)\phi_i - \bar{\bar{\phi}}_i^T(p_j^2M + p_jB + K)p_i\phi_i = 0 \quad . \qquad (26)$$

Combine terms:

$$(p_i - p_j)\bar{\bar{\phi}}_j^T[p_i p_j M - K]\phi_i = 0 \quad , \tag{27}$$

so that for $p_i \neq p_j$

$$\bar{\bar{\phi}}_j^T[p_i p_j M - K]\phi_i = 0 \quad . \tag{28}$$

Postmultiply Equation 25 by ϕ_i, and find that

$$-\bar{\bar{\phi}}_j^T K \phi_i = \bar{\bar{\phi}}_j^T(p_j^2 M + p_j B)\phi_i \quad . \tag{29}$$

Substitute this result into Equation 28:

$$(p_i + p_j)\bar{\bar{\phi}}_j^T M \phi_i + \bar{\bar{\phi}}_j^T B \phi_i = 0 \quad , \tag{30}$$

which is the desired result. Equation 30 is used in the sweeping algorithm. Note that Equation 30 is the analog of the mass orthogonality relationship (Equation 5 of Section 10.4.2.1) extended to complex unsymmetric matrices, including the presence of damping.

10.4.4.3 Proof of Convergence for the Iteration Algorithm

Consider the vector

$$\{x_n\} = \left\{\begin{array}{c} v_n \\ \hline u_n \end{array}\right\} \quad , \tag{31}$$

introduced in Equation 7. Expand $\{x_n\}$ and $\{x_{n-1}\}$ in terms of eigenvectors

$$x_n = \sum_i \alpha_{i,n}\phi_i \quad , \tag{32}$$

$$x_{n-1} = \sum_i \alpha_{i,n-1}\phi_i \quad . \tag{33}$$

Then, substituting into Equation 8 and premultiplying by $\bar{\bar{\phi}}_j^T$ which is the left eigenvector satisfying Equation 20,

$$\bar{\bar{\phi}}_j^T A \sum_i \alpha_{i,n}\phi_i = \frac{1}{C_n}\sum_i \alpha_{i,n-1}\bar{\bar{\phi}}_j^T\phi_i \quad , \tag{34}$$

or, using Equation 18,

$$\sum_i \alpha_{i,n}\Lambda_i\bar{\bar{\phi}}_j^T\phi_i = \frac{1}{C_n}\sum_i \alpha_{i,n-1}\bar{\bar{\phi}}_j^T\phi_i \quad , \tag{35}$$

and using the orthogonality property, Equation 23,

$$\alpha_{j,n} = \frac{1}{\Lambda_j C_n}\alpha_{j,n-1} \quad . \tag{36}$$

Thus the eigenvector with lowest value of Λ_j (i.e., with eigenvalue closest to the shift point) increases more rapidly than the others. The above proof of convergence is curious in that it requires the postulation of a second eigenvalue problem. It would be more satisfying if such were not required.

A more serious objection to the proof of convergence is the assumption that the mass matrix, [M], is nonsingular, which assumption is required in the reduction of the eigenvalue problem to standard form (see Section 10.4.4.1). The objection may be resolved by examining what happens as [M] approaches singularity. What happens, in fact, is that one (or more if the defect of [M] is greater than one) of the eigenvalues approaches infinity. Convergence to the corresponding eigenvector from any (finite) shift point is impossible. Otherwise the problem is not exceptional, and it may be concluded that convergence to finite eigenvalues will exist even in the limit when [M] is singular.

It will be noted that the rate of convergence of eigenvectors is the same as the rate of convergence in the real eigenvalue problem (see Equation 12 of Section 10.4.2.1). Thus arguments concerning the rate of convergence of eigenvectors in the real eigenvalue problem also apply to the complex eigenvalue problem. In particular, the test used in the decision to change the shift point is the same.

10.4.4.4 Sweeping of Previously Found Eigenvectors

Before each iteration, components of previously found eigenvectors are removed from the vector $\{x_n\} = \begin{Bmatrix} v_n \\ \hdashline u_n \end{Bmatrix}$ in order to permit convergence to a new eigenvector. The procedure is as follows. Represent $\{x_n\}$ as

$$\left\{\begin{array}{c} v_n \\ \hline u_n \end{array}\right\} = \sum_{j=1}^{m} \alpha_{j,n} \left\{\begin{array}{c} p_j\phi_j \\ \hline \phi_j \end{array}\right\} + \left\{\begin{array}{c} v_n \\ \hline u_n \end{array}\right\}_p \quad , \tag{37}$$

where ϕ_j is a previously accepted approximation to an eigenvector, where the sum extends over all m previously extracted eigenvectors, and where the last term on the right is the "purified" trial vector. The problem is to evaluate $\alpha_{j,n}$. For this purpose it is helpful to form the scalar matrix product

$$S_{jn} = \bar{\bar{\phi}}_j^T [p_j M u_n + M v_n + B u_n] \quad , \tag{38}$$

where $\bar{\bar{\phi}}_j$ is a previously accepted left eigenvector. Expanding u_n and v_n in terms of eigenvectors as in Equation 37, except that m ranges over all eigenvalues,

$$S_{jn} = \bar{\bar{\phi}}_j^T \sum_{i=1}^{m} \alpha_{i,n} \Big((p_i + p_j) M \phi_i + B \phi_i \Big) \quad , \tag{39}$$

or using the orthogonality relationship, Equation 30,

$$S_{jn} = \alpha_{j,n} \bar{\bar{\phi}}_j^T (2p_j M + B) \phi_j \quad . \tag{40}$$

Thus, comparing Equations 38 and 40,

$$\alpha_{j,n} = \frac{\bar{\bar{\phi}}_j^T [p_j M u_n + M v_n + B u_n]}{\bar{\bar{\phi}}_j^T (2p_j M + B) \phi_j} \quad . \tag{41}$$

A subtle point is the fact that the same factor $\alpha_{j,n}$ may be used for both u_n and v_n as indicated in Equation 37. The reasons are that it is valid, under most circumstances, to regard the union of u_n and v_n as an independent vector (x_n), and that the dimension of the vector space for x_n is equal to the number of eigenvalues. Thus the independent eigenvectors $\left\{\begin{array}{c} p_j\phi_j \\ \hline \phi_j \end{array}\right\}$ form a basis for the vector space of x_n except for the circumstance described below.

THE INVERSE POWER METHOD WITH SHIFTS

When the damping matrix, [B], is null, the orthogonality condition, Equation 30, is identical in form to the orthogonality condition for real eigenvalues, $\bar{\bar{\phi}}_j^T M \phi_i = 0$. As a result, considering only the bottom half of Equation 37, the factors $\alpha_{j,n}$ may be calculated from

$$\alpha_{j,n} = \frac{\bar{\bar{\phi}}_j^T M u_n}{\bar{\bar{\phi}}_j^T M \, \phi_j} \quad , \tag{42}$$

as well as from Equation 41. Since v_n is not rigidly dependent on u_n, Equations 41 and 42 will, in general, give different values of $\alpha_{j,n}$. Thus, if $\boxed{B}$ is null, the assumption that $\alpha_{j,n}$ may be used for both u_n and v_n is not valid. For this case Equation 42 is used to evaluate $(u_n)_p$; $(v_n)_p$ is obtained by substitution of $(u_n)_p$ into the basic iteration algorithm, Equation 14,

$$(v_n)_p = \lambda_o \, (u_n)_p + \frac{1}{C_n} (u_{n-1})_p \tag{43}$$

The left eigenvector $\bar{\bar{\phi}}_j$ required in Equations 41 and 42 is evaluated after the corresponding right eigenvector, ϕ_j, and the eigenvalue, p_j, have been found. The procedure is to form the lower and upper triangular factors $[L_\ell]$ and $[U_\ell]$ of the dynamic matrix at the accepted value of the eigenvalue,

$$[L_\ell][U_\ell] = [M \, p_j^2 + B \, p_j + K] \quad , \tag{44}$$

and then to solve the equation

$$[M^T p_j^2 + B^T p_j + K]\{\bar{\bar{\phi}}_j\} \equiv [U_\ell]^T [L_\ell]^T \{\bar{\bar{\phi}}_j\} = \{F\} \quad , \tag{45}$$

for $\{\bar{\bar{\phi}}_j\}$ by forward and backward substitution (see Section 2.3). The excitation vector, {F}, may be specified arbitrarily since $[U_\ell]$ is nearly singular (see Section 10.3.8).

The above procedure is similar to that used in finding eigenvectors by the determinant method (see Section 10.3). It has the disadvantage that it requires an additional triangular decomposition of the dynamic matrix for each eigenvalue extracted in order to evaluate the left eigenvector. It is, however, more certain to obtain the correct left eigenvector than any other method that has been examined, and it is relatively efficient for problems with narrow bandwidths.

10.4.4.5 Convergence and Shift Criteria

The convergence and shift criteria for real eigenvalue analysis developed in Sections 10.4.2.2 and 10.4.2.3, and summarized in Equations 9 to 24 of Section 10.4.3, are used with slight modification in complex eigenvalue analysis. It should be noted that the ambiguity with respect to the sign of α_n noted in Equation 24 of Section 10.4.2.2 is still present and that it is removed by comparing the phases of selected components of u_n and u_{n-1}.

The derivation of formulas for the convergence parameters, Λ_1, Λ_2, and δ_n, uses the mass orthogonality property of eigenvectors. Since the mass orthogonality property, $\phi_j^T M \phi_i = 0$ for $i \neq j$, does not apply in complex eigenvalue analysis, these formulas require revision.

The orthogonality condition for complex eigenvalue analysis, as derived in Section 10.4.4.2, is

$$\bar{\bar{\phi}}_j^T[(p_i + p_j)M + B]\phi_i = 0 , \quad i \neq j . \tag{46}$$

It cannot be used directly in the convergence criteria because we have no advance knowledge of the left eigenvectors, $\bar{\bar{\phi}}_j^T$. For the special case of symmetric matrices, however, the left eigenvectors are equal to the right eigenvectors. For use in the general case, define

$$\delta_{ij} = \phi_i^T\left(M + \frac{1}{p_{1,n-1} + p_{2,n-1}} B\right)\phi_j , \tag{47}$$

which is assumed to be normalized to unity for $i = j$. The parameters $p_{1,n-1}$ and $p_{2,n-1}$ are the current estimates of p_1 and p_2. Also define

$$A_{m,n} = u_m^T\left(M + \frac{1}{p_{1,n-1} + p_{2,n-1}} B\right)u_n . \tag{48}$$

Then, substituting for u_m and u_n from Equations 19 and 20 on Page 10.4-5, and using Equation 47, we find that

$$A_{n,n} = \alpha_n^2 \left(1 + \delta_n + \sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) , \tag{49}$$

and

$$A_{n-1,n-1} = \alpha_{n-1}^2\left(1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n + \left(\frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) . \tag{50}$$

Substituting from Equations 49 and 50 into Equation 36, a good approximation to the lowest eigenvalue is

$$\Lambda_1 = \frac{1}{C_n}\left(\frac{A_{n-1,n-1}}{A_{n,n}}\right)^{1/2}\left(\frac{1 + \delta_n + \sqrt{\delta_n}\,(\delta_{12} + \delta_{21})}{1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n + \frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})}\right)^{1/2} \quad . \tag{51}$$

The error in replacing the last bracketed term by unity is seen to be, since $\delta_n \ll \sqrt{\delta_n}$, of the order of $\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})$.

The normalized difference of successive trial vectors is, by comparison with Equation 28 of Section 10.4.2.2,

$$\delta u_n = \frac{u_n}{(A_{n,n})^{1/2}} - \frac{u_{n-1}}{(A_{n-1,n-1})^{1/2}} \quad . \tag{52}$$

Substituting from Equations 49 and 50, and assuming $\delta_n \ll \sqrt{\delta_n}\,(\delta_{12} + \delta_{21}) \ll 1$,

$$\delta u_n \simeq \frac{u_n}{\alpha_n}\left[1 - \frac{1}{2}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right] - \frac{u_{n-1}}{\alpha_{n-1}}\left[1 - \frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right] \quad , \tag{53}$$

and, using Equations 19 and 20 of Section 10.4.2.2

$$\begin{aligned}\delta u_n &= \left(\phi_1 + \sqrt{\delta_n}\,\phi_2\right)\left(1 - \frac{1}{2}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) \\ &\quad - \left(\phi_1 + \frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,\phi_2\right)\left(1 - \frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) \\ &\simeq \left(1 - \frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\left(\phi_2 - \frac{1}{2}(\delta_{12} + \delta_{21})\phi_1\right) \quad .\end{aligned} \tag{54}$$

Similarly

$$
\begin{aligned}
\delta u_{n-1} &\simeq \frac{u_{n-1}}{\alpha_{n-1}}\left[1-\frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right]-\frac{u_{n-2}}{\alpha_{n-2}}\left[1-\frac{1}{2}\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right] \\
&\simeq \left(\phi_1+\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,\phi_2\right)\left(1-\frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right) \\
&\quad -\left(\phi_1+\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,\phi_2\right)\left(1-\frac{1}{2}\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right) \\
&\simeq \frac{\Lambda_2}{\Lambda_1}\left(1-\frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\left(\phi_2-\frac{1}{2}(\delta_{12}+\delta_{21})\phi_1\right)
\end{aligned}
\qquad (55)
$$

Therefore, using Equation 47,

$$
\begin{aligned}
\Delta_{n,n} &= \delta u_n^T\left(M+\frac{1}{p_{1,n-1}+p_{2,n-1}}B\right)\delta u_n \\
&= \left(1-\frac{\Lambda_2}{\Lambda_1}\right)^2\delta_n\left[1+\frac{1}{4}(\delta_{12}+\delta_{21})^2-\frac{1}{2}(\delta_{12}+\delta_{21})^2\right],
\end{aligned}
\qquad (56)
$$

and

$$
\begin{aligned}
\Delta_{n,n-1} &= \delta u_n^T\left(M+\frac{1}{p_{1,n-1}+p_{2,n-1}}B\right)\delta u_{n-1} \\
&= \left(\frac{\Lambda_2}{\Lambda_1}\right)\left(1-\frac{\Lambda_2}{\Lambda_1}\right)^2\delta_n\left[1+\frac{1}{4}(\delta_{12}+\delta_{21})^2-\frac{1}{2}(\delta_{12}+\delta_{21})^2\right] \quad .
\end{aligned}
\qquad (57)
$$

Dividing Equation 56 by Equation 57, it is seen that

$$
\frac{\Lambda_2}{\Lambda_1}=\frac{\Delta_{n,n-1}}{\Delta_{n,n}} \quad . \qquad (58)
$$

From Equation 56,

$$
\delta_n=\frac{\Delta_{n,n}}{\left(1-\frac{\Lambda_2}{\Lambda_1}\right)^2\left(1-\frac{1}{4}(\delta_{12}+\delta_{21})^2\right)} \quad . \qquad (59)
$$

Since we do not know what δ_{12} and δ_{21} are, we will assume that they are zero. It is seen that the resulting approximation to δ_n is accurate only if $(\delta_{12} + \delta_{21})/2 << 1$. If this condition is not satisfied, δ_n, which is used to estimate the degree of convergence, may have a large error. Convergence itself, however, is guaranteed by Equation 36. The only dangers are that the algorithm will stop iterating before satisfactory convergence is achieved, or that it will iterate too long.

10.4.4.6 Search Procedures

In extracting complex eigenvalues, rectangular search regions are set up in the complex plane as shown in Figure 4. It is intended that all eigenvalues within the search regions be extracted (as limited by the desired maximum number of roots). There may be any number of search regions, and the search regions may overlap. The user establishes search regions by means of the coordinates of end points (A_j, B_j) and the width of the region (ℓ_j). The user estimates the number of roots (N_j) in each search region. For problems with real coefficients, all roots are either real or they occur in conjugate complex pairs, and it is inefficient for the user to specify regions in the lower half of the complex plane since the existence of conjugate pairs is assumed. The user will not, however, be prevented from specifying a search in the lower half of the complex plane since it is useful, in many cases, for a search region to include the origin or a band along the real axis.

The tasks performed by NASTRAN in the search procedure are as follows:

1. Divide region I into square subregions and place a starting point at the center of each subregion. The side length of the square subregion is set equal to ℓ_1, and the number of subregions is selected to be

$$\left|\frac{B_1 - A_1}{\ell_1}\right| \leq N_r < \left|\frac{B_1 - A_1}{\ell_1}\right| + 1 \quad . \tag{59}$$

   The sum of the subregions redefines the search region as shown in Figure 5. The redefined search region is symmetrical with respect to the center of the original search region.

2. Select the starting point that is closest to the origin as the first starting point. Select the next closest starting point as the second starting point, etc.

3. Terminate search from any starting point when a root is found outside of a circle passing through the corners of its square, according to rules set forth in Section 10.4.2.5.

4. When the search in region 1 has been completed, proceed to regions 2, 3, etc., in order. Sweep out all previously extracted eigenvectors from whatever region.

10.4.5 Summary of Procedures for Complex Eigenvalue Analysis

The procedures for complex eigenvalue analysis are very similar, and in many instances identical, to those for real eigenvalue analysis. The only major change in the flow diagrams for real eigenvalue analysis (Figures 1, 2, and 3) is the amendment of Figure 2 to include the calculation of the left eigenvector after passage of the convergence tests. Changes in procedure are indicated below according to the numbered blocks in the flow diagrams.

1. Compute Distribution of Starting Points

See Section 10.4.4.6.

2. Select a Starting Point

See Section 10.4.4.6.

4.1 Decompose Dynamic Matrix

The matrix to be decomposed is

$$[\lambda_o^2 M + \lambda_o B + K] \quad , \tag{1}$$

which is, in general, complex and nonsymmetric and where λ_o is either a starting point, λ_s, a moved starting point, λ_s', or a shifted shift point, λ_o'. Double-precision arithmetic with partial pivoting is used.

4.2 One Vector Iteration (see Sections 10.4.4.1 and 10.4.4.4)

The iteration algorithm is, for $B \neq 0$,

$$[\lambda_o^2 M + \lambda_o B + K]\{w_n\} = -[B + \lambda_o M]\{u_{n-1}\} - [M]\{v_{n-1}\} \quad , \tag{2}$$

$$\{\overline{u}_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{3}$$

$$\{\overline{v}_n\} = \lambda_o\{\overline{u}_n\} + \frac{1}{C_n}\{u_{n-1}\} \quad , \tag{4}$$

$$\{u_n\} = \{\bar{u}_n\} - \sum_j \alpha_j\{\phi_j\} \quad , \tag{5}$$

$$\{v_n\} = \{\bar{v}_n\} - \sum_j \alpha_j p_j\{\phi_j\} \quad , \tag{6}$$

where

$$\alpha_j = \frac{\{\bar{\phi}_j\}^T\{p_j M\bar{u}_n + M\bar{v}_n + B\bar{u}_n\}}{\{\bar{\phi}_j\}^T[2p_j M + B]\{\phi_j\}} \tag{7}$$

C_n = largest element (in magnitude) of $\{w_n\}$,

$\{\phi_j\}$ = previously found eigenvector ,

$\{\bar{\phi}_j\}$ = previously found left eigenvector ,

P_j = previously found eigenvalue ,

λ_o = shift point.

The sweeping operation, Equations 5 and 6, is also applied to starting vectors. The sum on (j) extends over all previously found eigenvalues.

For the special case, [B] = 0, Equation 6 is replaced by

$$\{v_n\} = \lambda_o\{u_n\} + \frac{1}{C_n}\{u_{n-1}\} \quad , \tag{8}$$

and Equation 7 is replaced by

$$\alpha_j = \frac{\{\bar{\phi}_j\}^T[M]\{\bar{u}_n\}}{\{\bar{\phi}_j\}^T[M]\{\phi_j\}} \quad . \tag{9}$$

4.3 Convergence Tests

The convergence tests are identical to those for real eigenvalue analysis except that $\{F_n\}$ is formed in the following manner

$$\{F_n\} = \left[M + \frac{1}{2\lambda_0 + \Lambda_{1,n-1} + \Lambda_{2,n-1}} B\right]\{u_n\} \qquad (9a)$$

where $\Lambda_{1,n-1}$ and $\Lambda_{2,n-1}$ are the estimates of Λ_1 and Λ_2 from the previous iteration.

4.8 Compute Left Eigenvector

Left eigenvectors satisfy the equation

$$[p_j^2 M^T + p_j B^T + K^T]\{\bar{\phi}_j\} = 0 \quad , \qquad (10)$$

where p_j is identically equal to the eigenvalue for the given problem, i.e.,

$$[p_j^2 M + p_j B + K]\{\phi_j\} = 0 \quad . \qquad (11)$$

See Section 10.4.4.4 for the method employed in calculating $\{\bar{\phi}_j\}$.

5. Have We Found All the Eigenvalues?

The answer is "yes" if the number of eigenvalues found equals the total number estimated to be in the problem. The number found should include conjugate eigenvalues for problems with real M, B, and K matrices. The total number estimated to be in the problem is

$$N_t = 2N_{eq} - N_{oM} \quad , \qquad (12)$$

for problems in which $[B] \neq 0$, and

$$N_t = 2N_{eq} - 2N_{oM} \quad , \qquad (13)$$

for problems in which $[B] = 0$ and where

N_{eq} = number of rows in dynamic matrix

N_{oM} = number of columns of the mass matrix that are null.

6. Is Eigenvalue Outside Range? (see Figure 5)

The test is:

$$|p_j - \lambda_s| > R_c \;:\; \text{Outside Range}, \qquad (14)$$

where R_c is the distance from the starting point λ_s to the corners of the starting point's search region, see Figure 5.

7. Select Next-to-Last Trial Vector as Starting Vector

   a. If there was no shift in finding the previous eigenvalue

$$\bar{u}_{i+1,o} = u_{i,N-1} \quad , \tag{15}$$

$$\bar{v}_{i+1,o} = v_{i,N-1} \quad . \tag{16}$$

   b. If there were one or more shifts: Select $\bar{u}_{i+1,o}$ and $\bar{v}_{i+1,o}$ equal to the last vector before the first shift.

8. Too Close to Starting Point? (see Section 10.4.2.4)

The criterion is

$$\varepsilon \left| \frac{R_c}{P_i - \lambda_s} \right| > \varepsilon_3 \quad : \quad \text{Too close} \tag{17}$$

where

ε = criterion specified by user for convergence of eigenvectors (Default value = 10^{-4})

P_i = eigenvalue just found

λ_s = starting point

R_c = range.

The value stored in the program for ε_3 is .05.

THE INVERSE POWER METHOD WITH SHIFTS

REFERENCE

1. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

ORIGINAL PAGE IS
OF POOR QUALITY

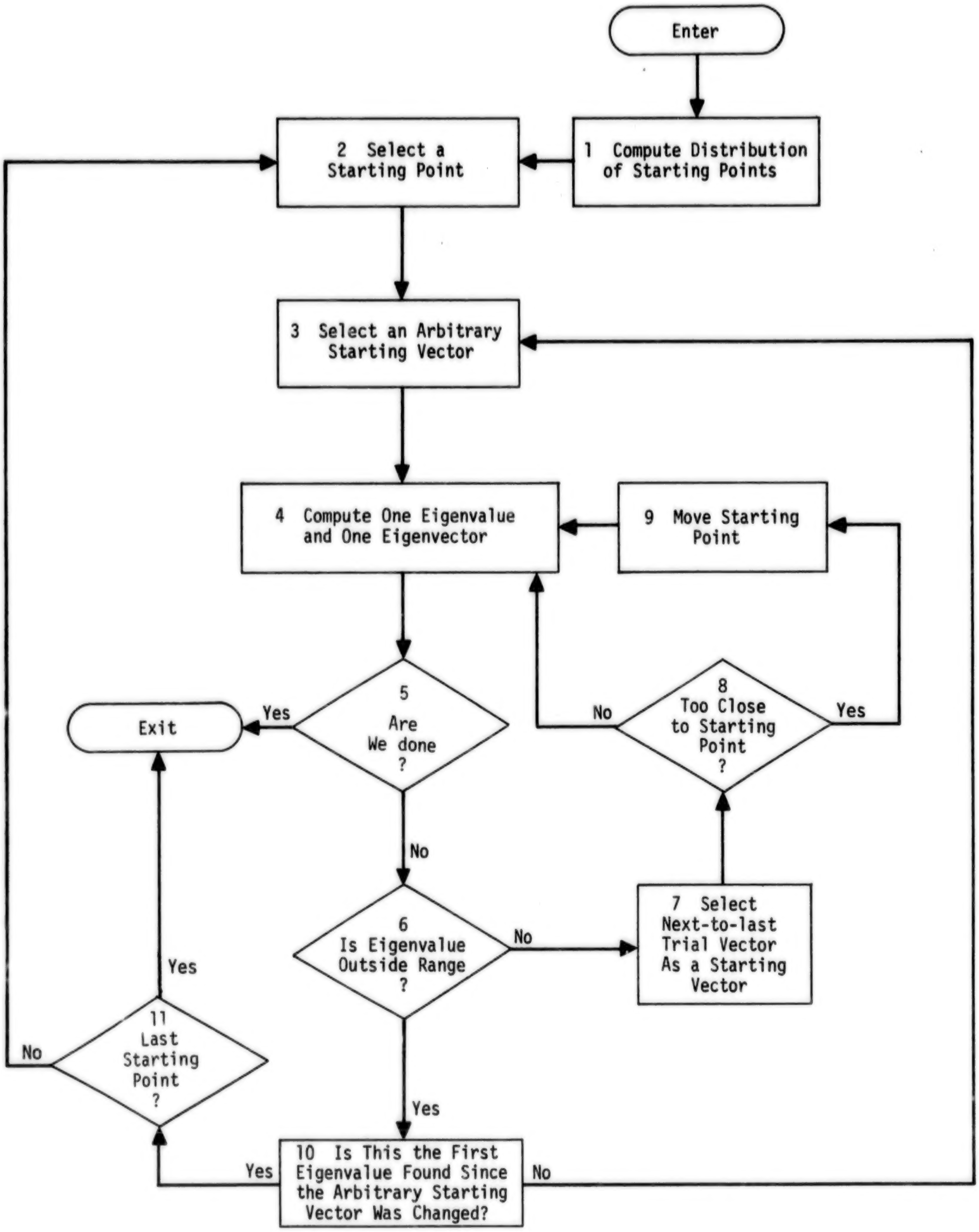
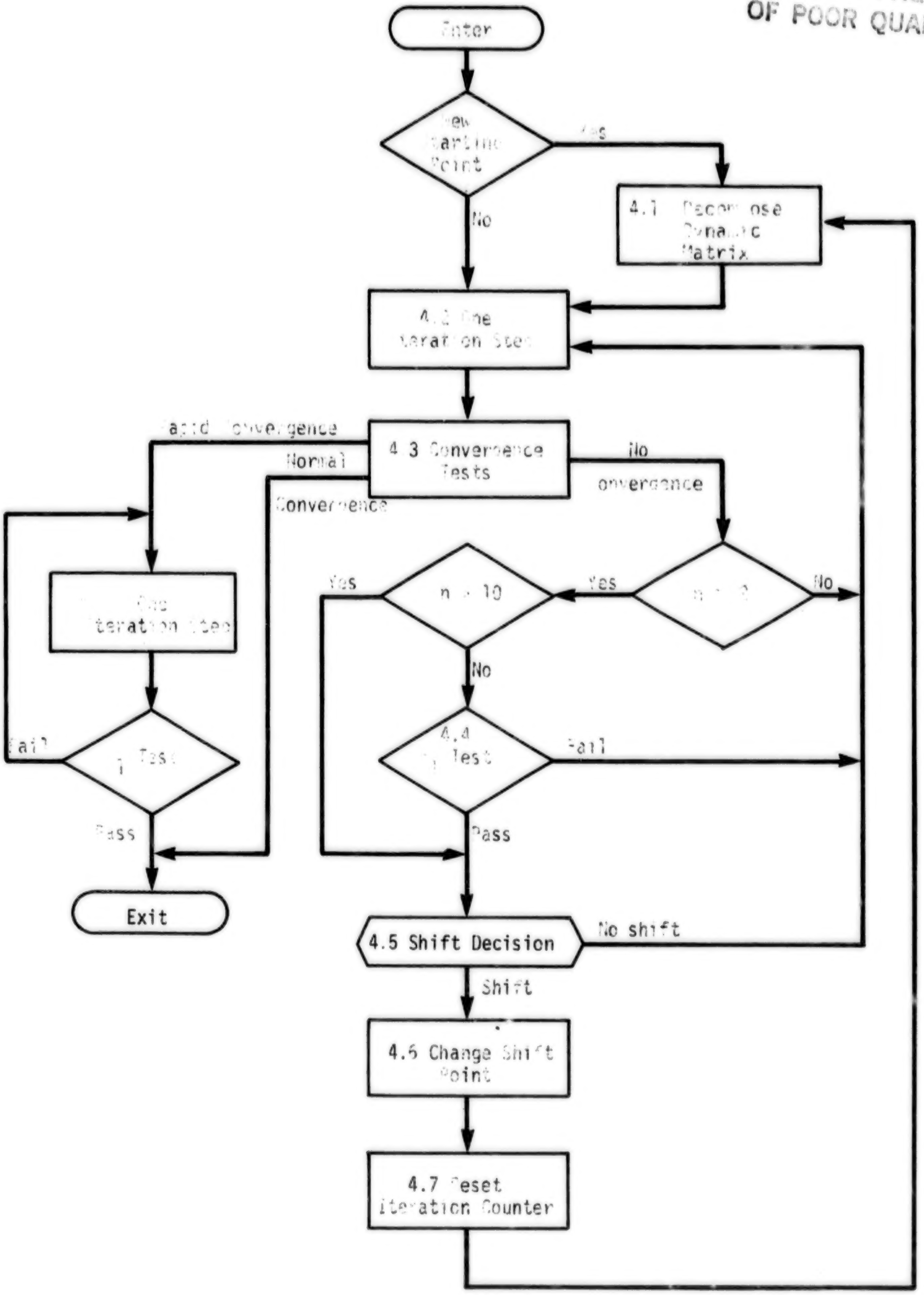

Enter
1 Compute Distribution of Starting Points
2 Select a Starting Point
3 Select an Arbitrary Starting Vector
4 Compute One Eigenvalue and One Eigenvector
9 Move Starting Point
5 Are We done ?
Yes
Exit
No
8 Too Close to Starting Point ?
No
Yes
6 Is Eigenvalue Outside Range ?
No
7 Select Next-to-last Trial Vector As a Starting Vector
Yes
11 Last Starting Point ?
No
Yes
10 Is This the First Eigenvalue Found Since the Arbitrary Starting Vector Was Changed?
Yes
No

Figure 1. Overall flow diagram for inverse power method with shifts.

THE INVERSE POWER METHOD WITH SHIFTS

ORIGINAL PAGE IS
OF POOR QUALITY



Enter
New Starting Point
Yes
No
4.1 Decompose Dynamic Matrix
4.2 One Iteration Step
Rapid Convergence
Normal Convergence
4.3 Convergence Tests
No Convergence
One Iteration Step
Fail
Test
Pass
Exit
Yes
n > 10
Yes
n > 2
No
No
4.4 Test
Fail
Pass
4.5 Shift Decision
No shift
Shift
4.6 Change Shift Point
4.7 Reset Iteration Counter

Figure 2. Flow diagram for block 4, compute one eigenvalue and one eigenvector.

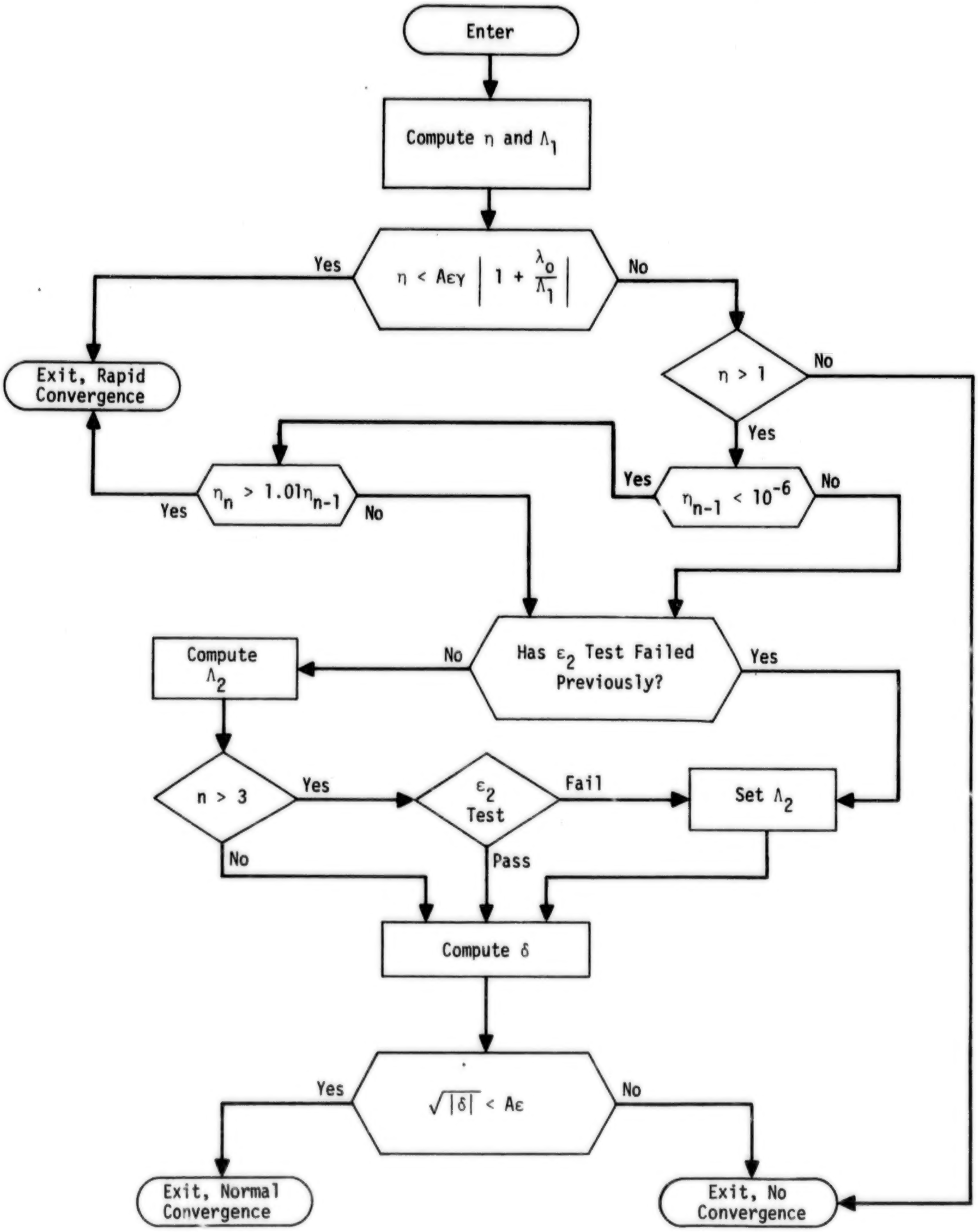

Enter
Compute η and Λ_1
$\eta < A\varepsilon\gamma \left| 1 + \frac{\lambda_o}{\Lambda_1} \right|$
Yes
No
Exit, Rapid Convergence
η > 1
No
Yes
$\eta_n > 1.01\eta_{n-1}$
Yes
No
$\eta_{n-1} < 10^{-6}$
Yes
No
Has ε_2 Test Failed Previously?
No
Yes
Compute Λ_2
n > 3
Yes
No
ε_2 Test
Fail
Pass
Set Λ_2
Compute δ
$\sqrt{|\delta|} < A\varepsilon$
Yes
No
Exit, Normal Convergence
Exit, No Convergence

Figure 3. Flow diagram for block 4.3, convergence tests.

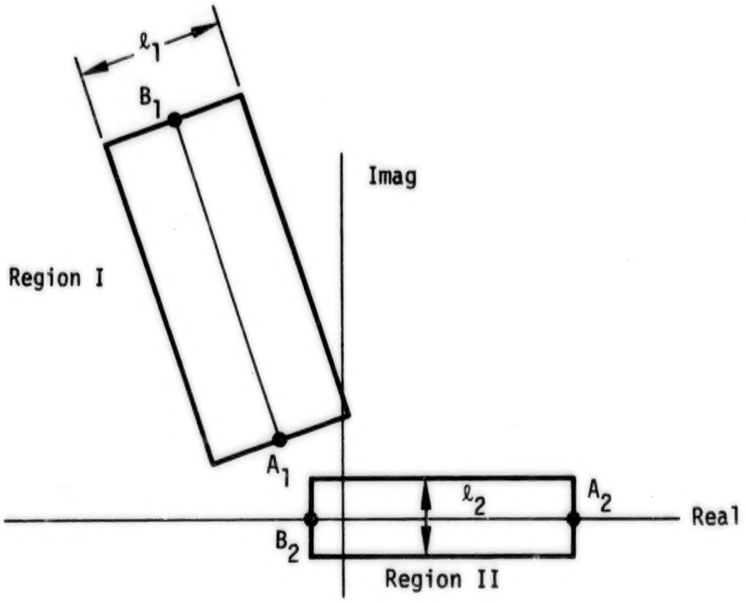

ℓ1
B1
Imag
Region I
A1
ℓ2
A2
Real
B2
Region II

Figure 4. Search regions.

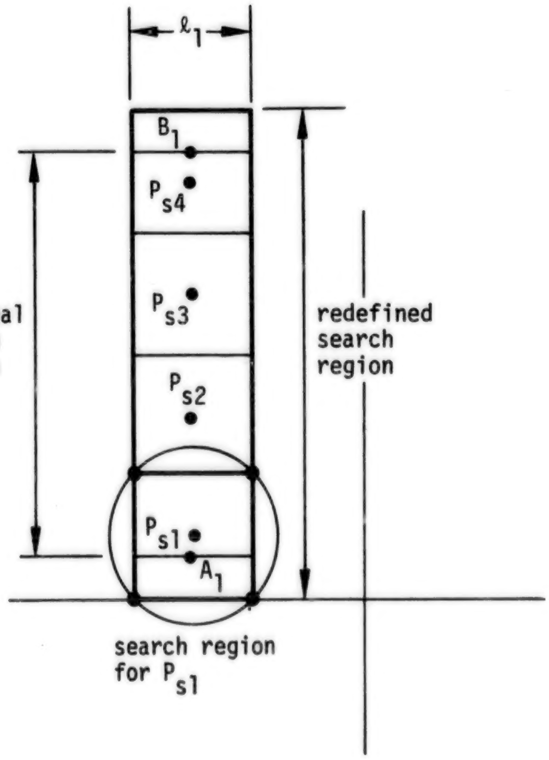

ℓ1
B1
Ps4
Ps3
Ps2
Ps1
A1
original search region
redefined search region
search region for Ps1

Figure 5. Distribution of starting points within a search region.

106

10.5 THE UPPER HESSENBERG METHOD

The upper Hessenberg method can be used for extraction of eigenvalues and eigenvectors from general, real or complex matrices. The techniques are more general than those employed in the Tridiagonal Method for Real Symmetric Matrices (Section 10.2). The fundamental reference is Wilkinson (Reference 1). The algorithms of Wilkinson for complex matrices have been automated by Funderlic and Rinzel in the subroutine ALLMAT (see Reference 2), which has been modified for NASTRAN.

The following outline is presented in the logical order of the calculations: Reduction to Canonical Form, Reduction to Upper Hessenberg Form, the QR Iteration, Convergence Criteria, Shifting, Deflation, and Eigenvectors.

10.5.1 Reduction to Canonical Form

The Upper Hessenberg Method requires that the eigenvalue problem be put in canonical form

$$[A - \lambda I]\{\phi\} = 0 \quad . \tag{1}$$

Matrix A is then reduced to the upper Hessenberg form through transformation techniques. The first step is the reduction to the form of Equation 1, which is performed automatically in module CEAD. Two cases are considered

1. $[Mp^2 + Bp + K]\{U\} = 0$ (The general form) (2)

$$A = \left[\begin{array}{c|c} 0 & I \\ \hline -M^{-1}K & -M^{-1}B \end{array}\right]$$

$p = \lambda$

U = The upper half of ϕ. The lower half of ϕ contains the velocity vector and is discarded.

2. $[Mp^2 + K]\{U\} = 0$ (B is not present) (3)

$$A = [-M^{-1}K]$$

$p = \sqrt{\lambda}$ with $Im(p) > 0$

$U = \phi$

PAGE 707 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

In Case 2, the decomposition of M is bypassed if it is an identity matrix. The reduction to canonical form requires that the matrix M must be nonsingular. The order of the problem is doubled when the matrix B is not null.

10.5.2 Reduction to Upper Hessenberg Form

If the given matrix is denoted as [A], it can be reduced to the upper Hessenberg matrix $[A_0]$ by using elementary stabilized transformations. The basic algorithm and two alternatives are given in Reference 1 (pp. 354-355). The selected method uses Equation 9.1 of Reference 1 (p. 355) in its reduction and it appears to be appropriate for the task. The total number of multiplications in the complete reduction is approximately $(5/6)n^3$, which is half the number in Householder's reduction and one-quarter the number in Givens' reduction.

10.5.3 The QR Iteration

The QR iteration of Francis (Reference 3) is defined by the relations (Reference 1, p. 515)

$$[A_0^{(s)}] = [Q^{(s)}][R^{(s)}] \quad , \tag{4}$$

$$[A_0^{(s+1)}] = [R^{(s)}][Q^{(s)}] \quad , \tag{5}$$

where $[Q^{(s)}]$ is the product of the (n-1) elementary unitary transformations necessary to reduce $[A_0^{(s)}]$ to the upper triangular form $[R^{(s)}]$ with positive real diagonal elements,

$$[Q^{(s)}] = [T_s^{(1)}][T_s^{(2)}] \ldots [T_s^{(n-2)}][T_s^{(n-1)}] \quad , \tag{6}$$

so that

$$[R^{(s)}] = [Q^{(s)}]^{-1}[A^{(s)}] \quad . \tag{7}$$

The transformation matrices $[T_s^{(j)}]$ are the Givens' rotations as discussed in Reference 1 (p. 239-240) (and in this manual, Section 10.2, in real form for real matrices) but in complex form for complex matrices. The iteration is continued until the n^{th} diagonal element $\left|a_{n,n-1}^{(s)}\right| < \varepsilon$, the convergence test, at which point the smallest eigenvalue $\lambda_1 = a_{n,n}^{(s)}$; if the convergence proceeds so that $\left|a_{n-1,n-2}^{(s)}\right| < \varepsilon$ before $\left|a_{n,n-1}^{(s)}\right| < \varepsilon$, the two smallest eigenvalues are the roots of

$$\begin{vmatrix} a_{n-1,n-1}^{(s)} - \lambda & a_{n-1,n}^{(s)} \\ a_{n,n-1}^{(s)} & a_{n,n}^{(s)} - \lambda \end{vmatrix} = 0 \quad . \tag{8}$$

The roots will be complex for complex matrices, and either real or complex conjugates for real matrices. Before each iteration, the subdiagonal elements should be tested and if some $|a_{i,i-1}^{(s)}| < \varepsilon$, the matrix should be split according to this occurrence, and the iteration continued with the lower main submatrix only.

10.5.4 Convergence Criteria

The convergence criteria suggested by Wilkinson (Reference 1, p. 526) is based on the Euclidean norm of the matrix $||A_0||_E$ and is

$$\varepsilon = 2^{-t} \, ||A_0||_E \qquad , \tag{9}$$

for floating-point calculations with mantissas of t binary bits. The Euclidean norm (Reference 1, p. 57) is found from

$$||A||_E^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \qquad . \tag{10}$$

Decimal equivalents of Equation 9 are used.

10.5.5 Shifting

Since the QR iteration converges to the smallest eigenvalue, the convergence can be accelerated by shifting, i.e., by subtracting scalar matrices from the original matrix. The matrix $[A_0^{(s)}]$ is replaced by the difference $[A_0^{(s)}] - k_s[I]$ after each iteration, in which k_s is an estimate of the eigenvalue. The shift eigenvalue k_s is that root of Equation 8, p_s or q_s, that makes $|a_{n,n}^{(s)} - p_s|$ or $|a_{n,n}^{(s)} - q_s|$ a minimum. The shifted algorithm then becomes (Reference 1, p. 524)

$$[A_0^{(s)}] - k_s[I] = [Q^{(s)}][R^{(s)}] \qquad , \tag{11}$$

and

$$[A_0^{(s+1)}] = k_s[I] + [R^{(s)}][Q^{(s)}] \qquad . \tag{12}$$

Equations 11 and 12 represent the algorithm for a shift to a single eigenvalue and is appropriate for a complex matrix or a real matrix with all real eigenvalues.

10.5.6 Deflation

When convergence to a single eigenvalue occurs, i.e., when $|a_{n,n-1}^{(s)}| < \varepsilon$, the Hessenberg matrix $[A_0]$ is deflated by elimination of its last row and column and the principal submatrix $[A_1]$ of order one less is the Hessenberg form for seeking the next eigenvalue. (Note: the subscript on A denotes the number of eigenvalues removed from $[A_0]$.) If convergence occurs to a pair of eigenvalues, i.e., $|a_{n-1,n-2}^{(s)}| < \varepsilon$, the matrix $[A_0]$ is deflated by deleting the last two rows and columns, and the principal submatrix $[A_2]$ of order two less becomes the basis for seeking the next eigenvalue. Each deflation removes either one or two eigenvalues depending on the two convergence tests.

10.5.7 Eigenvectors

The inverse power method with shifts (Reference 1, pp 626-628) converges rapidly to the eigenvector corresponding to each shift eigenvalue. This algorithm for real and complex matrices has been discussed thoroughly in Section 10.4. No further discussion is required here.

THE UPPER HESSENBERG METHOD

REFERENCES

1. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

2. Funderlic, R. E., and Rinzel, J., "ALLMAT, A FORTRAN IV Arbitrary Matrix Eigensystem Solver," SHARE Program Library, SDA No. 3441, 2 March 1966.

3. Francis, J. G. F., "The QR Transformation - a Unitary Analogue to the LR Transformation," Parts 1 and 2, The Computer Journal, Vol. 4, No. 3 (October 1961) and No. 4 (January 1962).

$$H_{12} = \sum_i \left(\frac{a_o{}^i + a_1{}^i p + a_2{}^i p^2}{b_o{}^i + b_1{}^i p + b_2{}^i p^2} \right) = \sum_i H^{(i)}(p) \quad , \tag{7}$$

which has the advantage of not requiring the numerator to be factored.

In this case auxiliary variables e_i are introduced such that

$$e_i = H^{(i)}(p) e_1 \quad , \quad i = 3, 4, 5, \text{ etc.}, \tag{8}$$

and

$$e_2 = \sum_i e_i \quad . \tag{9}$$

Equation 8 is treated as a set of differential equations, as before. Equation 9 is also treated as a differential equation where the only non-zero matrix elements are in the row of K corresponding to e_2.

In order to facilitate the treatment of control systems, NASTRAN includes an input data format for the specification of transfer functions in the form

$$u_e = \frac{1}{b_o + b_1 p + b_2 p^2} \sum_i \left(a_o{}^i + a_1{}^i p + a_2{}^i p^2 \right) u_i \quad , \tag{10}$$

which is interpreted by the program as a differential equation with terms in the rows of [M], [B], and [K] corresponding to u_e. u_e must be an extra point, but u_i can either be an extra point or any structural point contained in u_g.

Structural loads proportional to displacements at extra points are represented by adding terms in the stiffness matrix at the intersections of rows corresponding to the structural points and columns corresponding to the extra points. Direct input data cards are provided for this purpose. The superposition of all such terms, including those generated by transfer functions, is called the direct input stiffness matrix, $[K^2]$. The direct input mass and damping matrices are $[M^2]$ and $[B^2]$. Terms in the direct input matrices may refer to extra points, to structural points or to both. Since dynamic loads may also be applied to extra points, the means are available for the complete simulation of subsystems by means of extra points. Some applications to control system simulation are described in Section 14.2.

9.3.3 Direct Dynamic Matrix Assembly

The Direct Dynamic Matrix Assembler, GKAD, performs three main functions.

1. It decodes transfer function information and adds it to the direct input matrices $[M_{pp}{}^2]$, $[B_{pp}{}^2]$, and $[K_{pp}{}^2]$ as explained above.
2. It reduces the direct input matrices by the application of constraints and partitioning from the displacement set, u_p, (all physical points) to the u_d set. See Section 3.3.
3. It assembles the complete dynamic matrices $[M_{dd}]$, $[B_{dd}]$, and $[K_{dd}]$ from the direct input matrices and from the structural matrices generated in the statics portion of the program.

Direct input matrices are, at present, generated by transfer functions or they are supplied directly by the user. It is possible to modify the program so as to provide internal calculation of direct input matrices corresponding to particular aeroelastic and hydrodynamic environments.

The extra points are carried along in the application of constraints and partitioning to the direct input matrices, at the positions in u_p indicated in the grid point sequence list. For this purpose the multipoint constraint matrix $[G_m]$ (see Section 3.5.1), and the Guyan reduction matrix $[G_o]$, (see Section 3.5.4), are expanded to include the u_e coordinates. The corresponding columns are, of course, null.

The specific steps in the reduction as applied to $[K_{pp}{}^2]$ are:

1. Eliminate multipoint constraints.

$$[K_{nn}{}^2] = [\overline{K}_{nn}{}^2] + [K_{nm}{}^2][G_m] + [G_m]^T [K_{mn}{}^2] + [G_m]^T [K_{mm}{}^2][G_m] \quad , \qquad (11)$$

where the partitions of $[K_{pp}{}^2]$ are given by

$$[K_{pp}{}^2] = \left[\begin{array}{c|c} \overline{K}_{nn}{}^2 & K_{nm}{}^2 \\ \hline K_{mn}{}^2 & K_{mm}{}^2 \end{array}\right] \qquad (12)$$

(The u_e coordinates are included in the u_n coordinates.)

ORIGINAL PAGE IS
OF POOR QUALITY

2. Eliminate single-point constraints by extracting $[K_{ff}^2]$ from

$$[K_{nn}^2] = \left[\begin{array}{c|c} K_{ff}^2 & K_{fs}^2 \\ \hline K_{sf}^2 & K_{ss}^2 \end{array}\right] \quad . \tag{13}$$

(The u_e coordinates are included in the u_f coordinates.)

3. Eliminate the omitted coordinates, u_o, via the Guyan reduction

$$[K_{dd}^2] = [\overline{K}_{dd}^2] + [K_{do}^2][G_o] + [G_o]^T[K_{od}^2] + [G_o]^T[K_{oo}^2][G_o] \quad , \tag{14}$$

where the partitions of $[K_{ff}^2]$ are given by

$$[K_{ff}^2] = \left[\begin{array}{c|c} \overline{K}_{dd}^2 & K_{do}^2 \\ \hline K_{od}^2 & K_{oo}^2 \end{array}\right] \quad . \tag{15}$$

(The u_d set is the union of u_a and u_e.)

The direct input mass and damping matrices are treated in an identical manner. When the elements of the direct input matrices are complex numbers, which they may be for frequency response and complex eigenvalue analysis, the real and imaginary parts are reduced at the same time.

The final assembly of the dynamic matrices is performed as follows. The structural matrices $[K_{aa}]$, $[K_{aa}^4]$, $[M_{aa}]$, and $[B_{aa}]$ are expanded by the addition of zeroes in the rows and columns corresponding to extra points, to form $[K_{dd}^1]$, $[K_{dd}^4]$, $[M_{dd}^1]$, and $[B_{dd}^1]$. $[K_{aa}^4]$ is a structural damping matrix obtained by multiplying the stiffness matrix $[K_e]$ of an individual structural element by a damping factor, g_e, and combining the results for all such structural elements. Each element may have a different damping factor. $[B_{aa}]$ is a viscous damping matrix resulting from viscous rod elements (extension or twist) and viscous scalar dampers.

For frequency response analysis and complex eigenvalue analysis the complete dynamic matrices are

$$[K_{dd}] = (1 + ig)[K_{dd}^1] + [K_{dd}^2] + i[K_{dd}^4] \quad , \tag{16}$$

62

ORIGINAL PAGE IS
OF POOR QUALITY

$$[B_{dd}] = [B_{dd}{}^{1}] + [B_{dd}{}^{2}] \quad , \tag{17}$$

$$[M_{dd}] = [M_{dd}{}^{1}] + [M_{dd}{}^{2}] \quad . \tag{18}$$

$[K_{dd}{}^{1}]$ is multiplied by the factor $(1 + ig)$ to include uniform structural damping in cases where it is appropriate. The constant g is specified by the user.

For transient response analysis the complete dynamic matrices are

$$[K_{dd}] = [K_{dd}{}^{1}] + [K_{dd}{}^{2}] \quad , \tag{19}$$

$$[B_{dd}] = [B_{dd}{}^{1}] + [B_{dd}{}^{2}] + \frac{g}{\omega_3}[K_{dd}{}^{1}] + \frac{1}{\omega_4}[K_{dd}{}^{4}] \quad , \tag{20}$$

$$[M_{dd}] = [M_{dd}{}^{1}] + [M_{dd}{}^{2}] \quad . \tag{21}$$

The constant ω_3 is the radian frequency at which the term $\frac{g}{\omega_3}[K_{dd}{}^{1}]$ produces the same damping forces as the term $ig[K_{dd}{}^{1}]$ in frequency response analysis. The viscous damping forces are larger at higher frequencies and smaller at lower frequencies. A small value of g/ω_3 is frequently useful to ensure stability of higher modes in nonlinear transient analysis. The user selects the values of ω_3 and ω_4. If they are omitted the corresponding terms are ignored.

9.3.4 Modal Dynamic Matrix Assembly

The Modal Dynamic Matrix Assembler, GKAM, assembles the dynamic matrices $[M_{hh}]$, $[B_{hh}]$, and $[K_{hh}]$ in the general dynamic equation

$$[M_{hh}\, p^2 + B_{hh}\, p + K_{hh}]\{u_h\} = \{P_h\} \quad . \tag{22}$$

The modal analysis coordinates, u_h, are related to the corresponding physical coordinates, u_d, as follows. The set u_h is the union of the modal coordinates ξ_i and the extra points u_e

$$\left\{\begin{array}{c}\xi_i\\ \hline u_e\end{array}\right\} \longrightarrow \{u_h\} \quad . \tag{23}$$

603

while u_d is the union of u_a and u_e. The transformation between ξ_i and u_a is

$$\{u_a\} = [\phi_{ai}]\{\xi_i\} \quad , \tag{24}$$

where $[\phi_{ai}]$ is the matrix of eigenvectors obtained in real eigenvalue analysis, (see Section 9.2.1).

The transformation from u_h to u_d is obtained by augmenting $[\phi_{ai}]$ to include the extra points. Thus

$$\{u_d\} = [\phi_{dh}]\{u_h\} \quad , \tag{25}$$

where

$$[\phi_{dh}] = \left[\begin{array}{c|c} \phi_{ai} & 0 \\ \hline 0 & I \end{array}\right] \quad . \tag{26}$$

$[\phi_{dh}]$ is actually in this form because the components of u_e are placed last in u_h and also in u_d for modal analysis.

The dynamic system properties, exclusive of loads, included in dynamic analysis by the modal method are the direct input matrices $[M_{dd}{}^2]$, $[B_{dd}{}^2]$, and $[K_{dd}{}^2]$, obtained from the Direct Dynamic Matrix Assembler, and the modal matrices, $[m_i]$, $[b_i]$, and $[k_i]$, obtained from the Real Eigenvalue Analysis module. $[m_i]$ is the modal mass matrix defined by Equation 2 of Section 9.2.1, with off-diagonal terms (which should be zero anyway) omitted. The damping matrix $[b_i]$ and the stiffness matrix $[k_i]$ are obtained from $[m_i]$ by

$$[b_i] = [\omega_i \; g(\omega_i) m_i] \quad , \tag{27}$$

$$[k_i] = [\omega_i{}^2 \; m_i] \quad , \tag{28}$$

where ω_i is the radian frequency of the ith normal mode and $g(\omega_i)$ is a damping factor. $g(\omega_i)$ is obtained by interpolation of a table, $g(\omega)$, supplied by the user to represent the variation of structural damping with frequency.

It will be noted that the specification of damping properties for the modal method is somewhat less general than it is for the direct method in that viscous dampers $[B_{aa}]$ and nonuniform structural damping $[K_{aa}{}^4]$ are not used. The damping function $g(\omega)$ is regarded as an adequate substitute.

614

The complete dynamic matrices for the modal method of dynamic analysis are obtained from

$$[M_{hh}] = \left[\begin{array}{c|c} m_i & 0 \\ \hline 0 & 0 \end{array}\right] + [\phi_{dh}]^T [M_{dd}{}^2][\phi_{dh}] \quad , \tag{29}$$

$$[B_{hh}] = \left[\begin{array}{c|c} b_i & 0 \\ \hline 0 & 0 \end{array}\right] + [\phi_{dh}]^T[B_{dd}{}^2][\phi_{dh}] \quad , \tag{30}$$

$$[K_{hh}] = \left[\begin{array}{c|c} k_i & 0 \\ \hline 0 & 0 \end{array}\right] + [\phi_{dh}]^T[K_{dd}{}^2][\phi_{dh}] \quad . \tag{31}$$

The user can specify the number of modes, starting from zero frequency and including rigid body modes, or he can specify that all the modes within a prescribed frequency range be used.

A useful feature of NASTRAN is that the user can specify the structural properties of parts of the structure by means of scalar elements and constraints whose values are derived from substructure modes (see Section 14.1). The program treats such elements in the same manner as all other elements, so that their effects are included in $[K_{aa}]$, $[M_{aa}]$, and $[B_{aa}]$.

9.4 DYNAMIC DATA RECOVERY

The dynamic analysis modules for complex eigenvalue analysis, frequency response analysis, and transient response analysis produce either $\{u_d\}$ or $\{u_h\}$ as solution vectors, depending on whether a direct or a modal formulation has been employed. The transient response module also produces the first and second time derivatives of the solution vectors. The additional operations in the data recovery chain are indicated in Figure 1 of Section 9.1. The solution vector is partitioned, transformed, and augmented in various ways to obtain desired output quantities.

The main function of the Dynamic Data Recovery module (DDR), see Figure 1 of this section, is to produce a displacement vector, $\{u_a\}$, which can be processed by the static data recovery modules to produce the complete structural displacement vector, $\{u_g\}$, and the internal forces and stresses in selected structural elements (see Section 3.7). In the standard method of dynamic data recovery, which was the only method available prior to Level 16 of NASTRAN, the static data recovery modules process the solution vector for each mode, time step, or frequency as though it were a static solution vector, i.e., all calculations are repeated for every vector. Beginning with Level 16 of NASTRAN, a second method, called the Matrix Method of Dynamic Data Recovery, has been added for transient response and frequency response which is considerably more efficient when the number of time steps or frequencies exceeds the number of independent degrees of freedom. In this method a matrix is precomputed which relates the desired output quantities of whatever type directly to the independent degrees of freedom. Thus, in direct analysis,

$$\{u_j\} = [S_{jd}]\{u_d\} \quad , \tag{1}$$

and in modal analysis,

$$\{u_j\} = [S_{jh}]\{u_h\} \quad . \tag{2}$$

$[S_{jd}]$ and $[S_{jh}]$ are matrices of constant coefficients and $\{u_j\}$ is the vector of desired output quantities, which may include displacements, velocities, accelerations, internal forces, stresses, and forces of constraint. Note that the relationship of output quantities to independent degrees of freedom is linear so that nonlinear stress resultants, such as principal stresses, are excluded.

The matrix $[S_{jd}]$ is formed in two steps. First, the columns of an identity matrix of order $(d \times d)$ are processed sequentially by module SDR1, which recovers the degrees of freedom eliminated by constraints (see Section 3.7.1), to form a matrix $[u_{pd}]$ in which each column is the vector of grid point displacements, extra point displacements, and single point forces of constraint due to a unit value of one dynamic degree of freedom. Second, the columns of $[u_{pd}]$ are processed sequentially by module SDR2 (see Section 3.7.2) to form a matrix $[\sigma_{ed}]$ in which each column is the vector of user-requested stresses and internal forces in elements due to unit value of one dynamic degree of freedom. The union of $[\sigma_{ed}]$ with user-requested rows of $[u_{pd}]$ form the matrix $[S_{jd}]$.

The matrix $[S_{jh}]$ is formed in a similar way except that in the first step the identity matrix is replaced by $[\phi_{dh}]$, the augmented modal matrix (see Section 9.3.4).

If the number of dynamic degrees of freedom is less than the number of time steps or frequencies, the matrix method of dynamic data recovery will be more efficient than the standard method. The saving may be very great, particularly in the case of the modal formulation when the number of modes is small.

In the event that it is desired to print or plot each output quantity versus time or frequency, it is required that the matrix of output quantities, $[u_j]$, be transposed. This is actually performed by transposing the output data recovery matrix first, followed by the calculation

$$[u_j]^T = [u_d]^T[S_{jd}]^T \quad \text{or} \quad [u_j]^T = [u_h]^T[S_{jh}]^T \quad . \tag{3}$$

One of the two methods for multiplying $[A]^T[B]$ described on page 2.1-4 is used. Both methods avoid the transposition of $[u_h]$ or $[u_d]$.

In the mode acceleration method, which is only available as a subcase of the standard method (see Figure 1), the vector $\{u_a\}$ that describes unconstrained motions of the structure is replaced by another vector $\{u_a^a\}$ that will produce more accurate estimates of the stresses. The motivation stems from the fact that when modal analysis is employed, the modal vector $\{\xi_i\}$ that replaces $\{u_a\}$ is nearly always a smaller set from which higher modes have been omitted. The influence of the higher modes can be approximated by observing that their response to low frequency dynamic excitation is almost purely static. Thus, the inertia and damping forces on the structure contain very little contribution from the higher modes. In the mode acceleration

617

method, the inertia and damping forces are computed from the modal solution. These forces are then added to the applied forces and are used to obtain a more accurate displacement vector for the structure by static analysis. The details of the computation are as follows.

When the general dynamic system is formed using modes, the vector $\{u_h\}$ used in the analysis contains the modal coordinates $\{\xi_i\}$ and the extra degrees of freedom $\{u_e\}$. The system from which $\{\xi_i\}$ is abstracted consists of the stiffness matrix $[K_{aa}]$, the mass matrix $[M_{aa}]$, and modal damping elements, b_i. The presence of the modal damping elements is ignored in data recovery. The general system includes, in addition, direct input matrices $[M^2]$, $[B^2]$, and $[K^2]$ and loads $\{P_a\}$ and $\{P_e\}$ applied, respectively, to points $\{u_a\}$ and $\{u_e\}$.

In order to apply the mode acceleration method, the system that existed prior to the application of modal constraints is reconstructed, as follows:

$$[M_{dd}\, p^2 + B_{dd}{}^2\, p + (K_{dd}{}^1 + K_{dd}{}^2)]\{\bar{u}_d\} \;=\; \{P_d\} \quad . \tag{4}$$

$\{\bar{u}_d\}$ differs from $\{u_d\}$, the solution vector actually obtained, due to the presence of higher modes. Note the presence of $[B_{dd}{}^2]$ from Equation 30 of Section 9.3.4.

The elastic properties of the structure are represented by $[K_{dd}{}^1]$ which is the structural stiffness matrix $[K_{aa}]$ augmented by zeroes in the rows and columns corresponding to the extra points, u_e, (see Section 9.3.3). $[K_{dd}{}^1]$ represents the complete structure as opposed to the modal stiffness matrix $[k_i]$ which ignores the higher deformation modes. The viewpoint is taken that the transformed solution vector, $\{u_d\} = [\phi_{dh}]\{u_h\}$, is adequate to evaluate the inertia (and damping) loads on the structure but that stresses derived from it could be improved.

The improved solution is the vector $\{u_d^a\}$ obtained by solving

$$[K_{dd}{}^1]\{u_d^a\} \;=\; \{P_d^a\} \quad , \tag{5}$$

where, comparing Equations 4 and 5

$$\{P_d^a\} \;=\; \{P_d\} - [M_{dd}\, p^2 + B_{dd}{}^2\, p + K_{dd}{}^2]\{u_d\} \quad . \tag{6}$$

The derivative operator p in Equation 6 is simply a number for complex eigenvalue and frequency response analysis. For transient analyses the velocity and acceleration vectors, $\{pu_d\}$ and $\{p^2\, u_d\}$, are contained in the output of the transient analysis module.

The procedure indicated in Equations 5 and 6 is modified in the program by the removal of extra points, u_e, and by the use of a special procedure for free bodies. First $\{P_d^a\}$ and $\{u_d^a\}$ are partitioned

$$\{P_d^a\} \;=\; \left\{\begin{array}{c} P_a^a \\ \hdashline P_e \end{array}\right\} \tag{7}$$

$$\{u_d^a\} = \left\{\begin{array}{c} u_a^a \\ \hline u_e \end{array}\right\} \quad . \tag{8}$$

Then, if there are no free body motions, $\{u_a^a\}$ is obtained by solving

$$[K_{aa}]\{u_a^a\} = \{P_a^a\} \quad . \tag{9}$$

by the standard procedures used in static analysis. Note that for complex eigenvalue extraction and for frequency response analysis $\{P_a^a\}$ may be complex, in which event the real and imaginary parts are considered as separate loading conditions. The operations of the static data recovery module are then applied separately to the real and imaginary parts of $\{u_a^a\}$.

If free body modes are present, $[K_{aa}]$ is singular, indicating the need for an alternate procedure. It is assumed that the motions at a set of fictitious reaction points, u_r, (see Section 3.5.5) are correctly given by the modal solution. The elastic deformations of the remaining points, u_ℓ, relative to u_r are obtained by solving

$$[K_{\ell\ell}]\{u_\ell^a\} = \{P_\ell^a\} \quad , \tag{10}$$

where $\{P_\ell^a\}$ is the appropriate partition of

$$\{P_a^a\} = \left\{\begin{array}{c} P_\ell^a \\ \hline P_r^a \end{array}\right\} \quad . \tag{11}$$

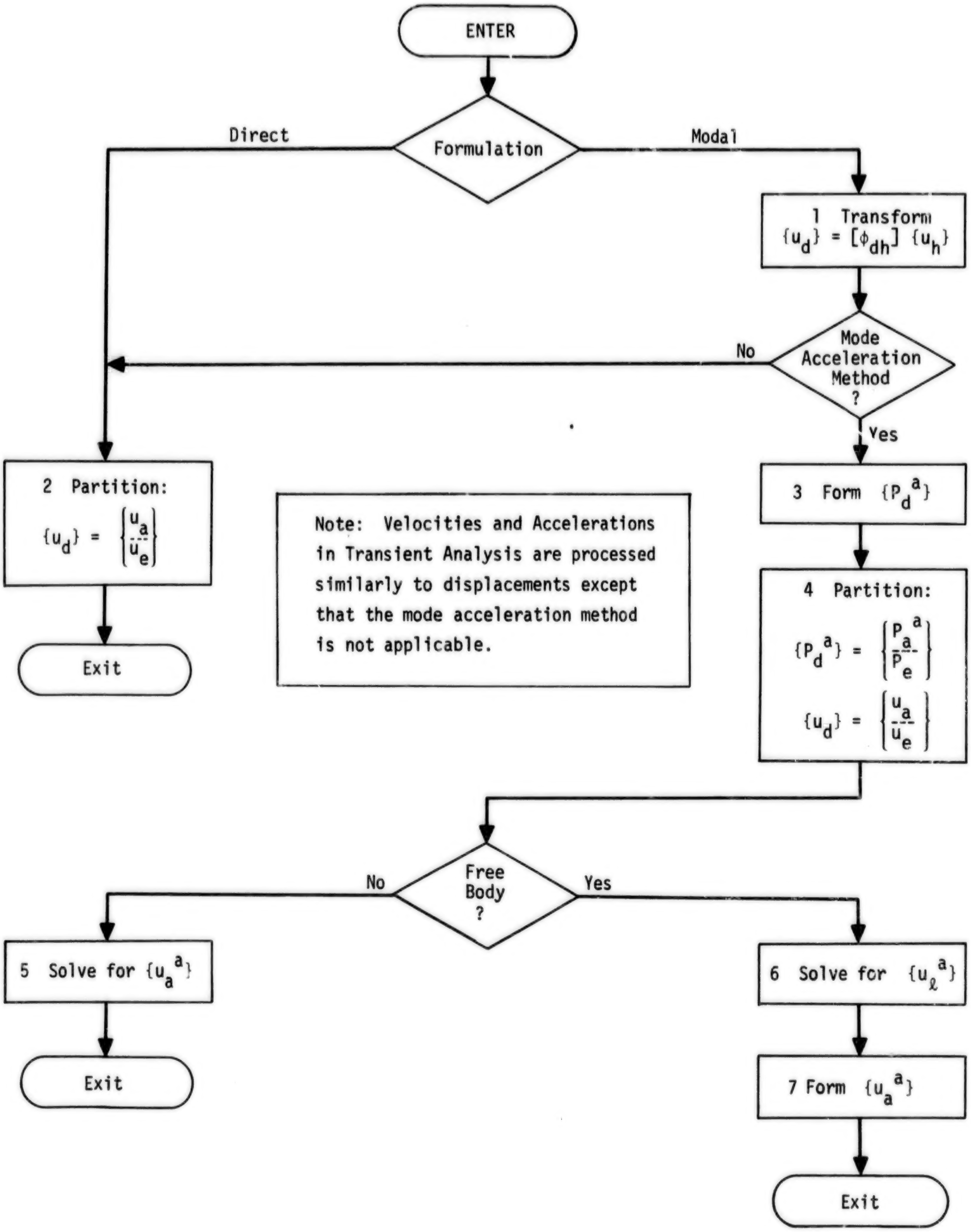
The improved solution vector is obtained from

$$\{u_a^a\} = \left\{\begin{array}{c} u_\ell^a + D\,u_r \\ \hline u_r \end{array}\right\} \quad , \tag{12}$$

where [D] is the matrix that evaluates the rigid body motions of the structure (see Section 3.5.5).

Although the mode acceleration technique will produce better stresses, it may be expensive, particularly in transient analyses involving many time steps. Its use must, therefore, be separately justified for each application.

ORIGINAL PAGE IS OF POOR QUALITY


ENTER
Formulation
Direct
Modal
1 Transform $\{u_d\} = [\phi_{dh}] \{u_h\}$
Mode Acceleration Method ?
No
Yes
2 Partition: $\{u_d\} = \left\{\frac{u_a}{u_e}\right\}$
Exit
3 Form $\{P_d^a\}$
4 Partition:
$\{P_d^a\} = \left\{\frac{P_a^a}{P_e}\right\}$
$\{u_d\} = \left\{\frac{u_a}{u_e}\right\}$
Free Body ?
No
Yes
5 Solve for $\{u_a^a\}$
Exit
6 Solve for $\{u_\ell^a\}$
7 Form $\{u_a^a\}$
Exit

Note: Velocities and Accelerations in Transient Analysis are processed similarly to displacements except that the mode acceleration method is not applicable.

Figure 1. Dynamic Data Recovery, Standard Method

10. EIGENVALUE EXTRACTION METHODS

10.1 INTRODUCTION

Four methods of eigenvalue extraction are provided with NASTRAN because no single method, or pair of methods, has been found that is satisfactory with respect to efficiency, reliability, and generality of application in all situations. We are living in a time of rather explosive growth, engendered by the digital computer, in the variety and capability of eigenvalue extraction routines. New methods will be added to NASTRAN and old methods will be improved as discoveries are made.

Most methods of algebraic eigenvalue extraction belong to one of two groups - transformation methods and "tracking" methods. In a transformation method the matrix of coefficients is first transformed, while preserving its eigenvalues, into a special form (diagonal, tridiagonal, or upper Hessenberg) from which eigenvalues may easily be extracted. In a "tracking" method the roots are extracted, one at a time, by iterative procedures applied to the original dynamic matrix . Two of the methods used in NASTRAN are transformation methods (Tridiagonal Method and Tridiagonal Reduction or FEER Method), and the other two (Determinant Method and Inverse Power Method with Shifts) are tracking methods.

In the Tridiagonal Method, the full matrix of coefficients is converted into a condensed form, requiring that the major share of the total effort be expended prior to the extraction of the first eigenvalue. Thus, the total effort is not strongly dependent on the number of eigenvalues that are extracted. However, in the Tridiagonal Reduction Method, the matrix is not only converted to a condensed (tridiagonal) form, but it is simultaneously reduced in size. The size of the reduced matrix is of the same order of magnitude as the number of modes requested by the user. Therefore, this method possesses a major efficiency as compared to the straightforward Tridiagonal Method, namely that the effort expended is proportional to the number of extracted eigenvalues. The computations required in the tracking methods are also linearly proportional to the number of extracted eigenvalues and as a result they tend to be more efficient when only a few eigenvalues are required.

The general characteristics of the methods used in NASTRAN are compared in Table 1. The Tridiagonal Method, due to restrictions on matrix form, is available only for the evaluation of the vibration modes of conservative systems (see Section 9.2) and not for buckling or complex eigenvalue analysis. The Tridiagonal Reduction (FEER) Method, however, accommodates both vibration modes and buckling analyses. The other two methods are available for all real and complex eigenvalue problems currently solved by NASTRAN. The Determinant Method would, in addition, be applicable to future problem types in which the coefficients of the dynamic matrices are general functions of the eigenvalue.

It may be noted, from Table 1, that a narrow bandwidth, as well as a small proportion of extracted roots, tends to favor the tracking methods and the Tridiagonal Reduction Method. An example of such a problem is the evaluation of the lowest few modes of a launch vehicle.

When several modes of a real eigenvalue problem are required, the Tridiagonal Reduction Method will probably be the most efficient one to use, since it is a finite, non-iterative process usually requiring only one initial eigenvalue shift. When almost all the modes are required, the Tridiagonal Method may exceed the efficiency of the Tridiagonal Reduction Method, particularly if the bandwidth is relatively large. It should be noted, however, that when the order of the coefficient matrix exceeds two or three hundred, the Tridiagonal Method will probably require prohibitively time-consuming out-of-core operations, even on computers with relatively large central memory capacities, thereby losing its efficiency. This disadvantage does not apply to the Tridiagonal Reduction Method which is more efficient for out-of-core operations.

The Determinant Method and the Inverse Power Method with Shifts have the same general characteristics with respect to all current NASTRAN problems. The Inverse Power Method is, however, more efficient except when the bandwidth is extremely narrow as, for example, in beam problems. The main advantage of including both methods is the redundancy that is provided in case one method should fail (as sometimes happens with any method of eigenvalue extraction).

ORIGINAL PAGE IS OF POOR QUALITY

Table 1. Comparison of Methods of Eigenvalue Extraction

Method Characteristics	Tridiagonal Method	Inverse Power Method With Shifts	Determinant Method	Tridiagonal Reduction Method
Most general form of matrix	$[A - pI]$	$[Mp^2 + Bp + K]$	$[A(p)]$	$[Mp^2 + K]$
Restrictions on matrix character	A real, sym., constant	M, B and K constant	None	M, K, real, sym., constant
Obtains eigenvalues in order	All at once	Nearest to shift point	(Usually) nearest to starting points	Vibration: Nearest to shift point Buckling: Lowest eigenvalue first
Takes advantage of bandwidth	No	Yes	Yes	Yes
Number of calculations, order of	$O(n^3)$	$O(nb^2E)$	$O(nb^2E)$	$O(n[b+E]^2)$

Note: n = number of equations

b = simi-bandwidth

E = number of eigenvalues extracted

10.2 THE TRIDIAGONAL METHOD

10.2.1 Introduction

The tridiagonal method is particularly effective for obtaining vibration modes when all, or a substantial fraction, of the eigenvalues and eigenvectors of a real symmetric matrix are desired. The general restrictions on the use of the method within NASTRAN are described in Section 10.1. The basic steps employed in the method are as follows. First, the eigenvalue problem,

$$[K_{aa} - \lambda M_{aa}]\{u_a\} = 0, \tag{1}$$

is transformed to the form

$$[A]\{x\} = \lambda\{x\}. \tag{2}$$

Second, [A] is converted, while preserving its eigenvalues, to a tridiagonal matrix by the Givens Method, Reference 1. Third, all the eigenvalues of the tridiagonal matrix are extracted using a modified Q-R algorithm. Fourth, the eigenvectors are computed over a given frequency range or for a given number of eigenvalues, and are converted to physical form. The transformation of the eigenvalue problem from Equation 1 to Equation 2 and the conversion of the eigenvectors to physical form are described in Section 9.2. A simplified flow diagram for the tridiagonal method is shown in Figure 1.

10.2.2 Tridiagonalization by the Givens Method

The most stable transformation methods of solution of eigenproblems for large symmetric matrices are based on the tridiagonalization techniques of Givens (Reference 1), Lanczos (Reference 2), and Householder (Reference 3). The Givens method, which is used by NASTRAN, depends on orthogonal transformations $[T]\,[A]\,[T]^T$ of a symmetric matrix [A]. An orthogonal transformation is one whose matrix [T] satisfies

$$[T][T]^T = [T]^T[T] = [I]. \tag{3}$$

The eigenvalues of a matrix are preserved under orthogonal transformation since

$$[T]\Big([A] - \lambda[I]\Big)[T]^T = [T][A][T]^T - \lambda[I]. \tag{4}$$

Consequently if det $\Big([A] - \lambda[I]\Big)$ vanishes, then det $\Big([T][A][T]^T - \lambda[I]\Big)$ also vanishes.

PAGE 626 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

The effect of a series of orthogonal transformations on the eigenvectors of a matrix is that of a succession of multiplications by orthogonal matrices. For if

$$[A]\{x\} = \lambda\{x\}, \tag{5}$$

and if $[T_1]$, $[T_2]$, ..., $[T_r]$ are orthogonal matrices, then

$$[T_r][T_{r-1}] \cdots [T_2][T_1][A]\{x\} = \lambda[T_r][T_{r-1}] \cdots [T_2][T_1]\{x\}. \tag{6}$$

Write $\{x\} = [T_1]^T [T_2]^T \cdots [T_{r-1}]^T [T_r]^T \{y\}$. Then

$$\begin{aligned} &[T_r][T_{r-1}] \cdots [T_2][T_1][A][T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T\{y\} \\ &\quad = \lambda[T_r][T_{r-1}] \cdots [T_2][T_1][T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T\{y\} \\ &\quad = \lambda\{y\}, \end{aligned} \tag{7}$$

where Equation 3 is applied repeatedly to obtain the final form. It follows that $\{y\}$ is an eigenvector of the transformed matrix

$$[T_r][T_{r-1}] \cdots [T_2][T_1][A][T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T,$$

and that $\{x\}$ may be obtained from $\{y\}$ by

$$\{x\} = [T_1]^T[T_2]^T \cdots [T_{r-1}]^T[T_r]^T \{y\}. \tag{8}$$

The Givens method uses orthogonal matrices $[T]$ which are identical with the unit matrix $[I]$ except for the four elements

$$\left.\begin{aligned} t_{i+1,\, i+1} &= t_{j,\, j} = \cos\theta_{i+1,\, j}\ , \\ t_{i+1,\, j} &= -t_{j,\, i+1} = \sin\theta_{i+1,\, j}\ . \end{aligned}\right\} \tag{9}$$

The orthogonal transformation $[T][A][T]^T$ leaves unchanged all the elements of $[A]$ except those in the $i+1^{st}$ and j^{th} rows and columns, the so-called plane of rotation. The four pivotal elements of the transformed matrix are

$$\left.\begin{aligned}\alpha_{i+1,\,i+1} &= a_{i+1,\,i+1}\cos^2\theta_{i+1,\,j} + a_{j,\,j}\sin^2\theta_{i+1,\,j} + a_{i+1,\,j}\sin 2\theta_{i+1,\,j}\,,\\ \alpha_{j,\,j} &= a_{i+1,\,i+1}\sin^2\theta_{i+1,\,j} + a_{j,\,j}\cos^2\theta_{i+1,\,j} - a_{i+1,\,j}\sin 2\theta_{i+1,\,j}\,,\\ \alpha_{i+1,\,j} = \alpha_{j,\,i+1} &= a_{i+1,\,j}\cos 2\theta_{i+1,\,j} - \frac{1}{2}\left(a_{i+1,\,i+1} - a_{j,\,j}\right)\sin 2\theta_{i+1,\,j}\,.\end{aligned}\right\}\tag{10}$$

where $a_{j,\,j}$ etc., are elements of the untransformed matrix. The other elements of the $i+1^{st}$ and j^{th} rows and columns of the transformed matrix are:

$$\left.\begin{aligned}\alpha_{i+1,\,s} = \alpha_{s,\,i+1} &= a_{i+1,\,s}\cos\theta_{i+1,\,j} + a_{j,\,s}\sin\theta_{i+1,\,j}\\ \alpha_{j,\,s} = \alpha_{s,\,j} &= -a_{i+1,\,s}\sin\theta_{i+1,\,j} + a_{j,\,s}\cos\theta_{i+1,\,j}\end{aligned}\right\}\tag{11}$$

In the Givens method $\theta_{i+1,\,j}$ is chosen such that $\alpha_{i,\,j}$ will vanish, which happens when

$$\tan\theta_{i+1,\,j} = a_{i,\,j}/a_{i+1,\,i}\,.\tag{12}$$

The calculation of $\theta_{i+1,\,j}$ followed by the orthogonal transformation

$$[A^{(m)}] = [T_m][A^{(m-1)}][T_m]^T,\tag{13}$$

is carried out for a sequence of iterations with $[A^{(0)}] = [A]$. The values of i used in Equations 9, 10, 11, and 12 are 1, 2, 3, ..., (n-2) and for each i, a set of n - i - 1 such transformations is performed, with j taking values i + 2, i + 3, ..., n before the next value of i is used. The result is that elements in the matrix positions (1,3), (1,4), ..., (1,n), (2,4), (2,5), ..., (2,n), ..., (n-2,n) are successively reduced to zero, together with their transposes, the (3,1), (4,1), ..., (n,n-2) elements. The set of transformations thus reduces the matrix to tridiagonal form.

NASTRAN employs a procedure introduced by Wilkinson (Reference 4) in which the Givens method is modified by grouping together the (n - i - 1) transformations which produce zeros in the i^{th} row and column. This procedure should not be confused with the Householder method, Reference 3, which annihilates a row and column at a time. The Wilkinson process is particularly advantageous when the matrix [A] is so large that the elements cannot all be left in core storage at one time; it requires only n - 2 transfers of the matrix to and from auxiliary storage instead of the (n-1)(n-2)/2 transfers required by the unmodified Givens method. The method requires 4n core storage locations for working space and these are divided into four groups of n storage locations each. The first (i-1) rows and columns play no part in the i^{th} major step. This step has five substeps:

1. The i^{th} row of [A] is transferred to the set of core storage locations in group 1.

2. The values of $\cos\theta_{i+1,\, i+2}$, $\sin\theta_{i+1,\, i+2}$ .... $\cos\theta_{i+1,\, n}$, $\sin\theta_{i+1,\, n}$ are computed successively from

$$\left.\begin{aligned} \cos\theta_{i+1,\, j} &= \frac{a^{(j-1)}_{i,\, i+1}}{\sqrt{\left(a^{(j-1)}_{i,\, i+1}\right)^2 + a^2_{i,\, j}}} \;, \\ \sin\theta_{i+1,\, j} &= \frac{a_{i,\, j}}{\sqrt{\left(a^{(j-1)}_{i,\, i+1}\right)^2 + a^2_{i,\, j}}} \;, \end{aligned}\right\} \qquad (14)$$

where the superscripted term is computed by

$$a^{(j-1)}_{i,\, i+1} = \sqrt{\left(a^{(j-2)}_{i,\, i+1}\right)^2 + a^2_{i,\, j-1}} \;, \qquad (15)$$

and the starting value for $j = i + 2$ is

$$a^{(i+1)}_{i,\, i+1} = a_{i,\, i+1} \;.$$

The $\cos\theta_{i+1,j}$ may be overwritten on the elements of the untransformed matrix, $a_{i,\, j}$, which are no longer required, and the $\sin\theta_{i+1,j}$ are stored in the group 2 storage locations.

3. The $i+1^{st}$ row of [A] is transferred to the group 3 storage locations. Only those elements on and above the diagonal are used in this and succeeding rows.

For k = i + 2, i + 3, ..., n in turn, the operations in substeps 4 and 5 are carried out.

4. The k^{th} row of [A] is transferred to the group 4 storage locations. The elements $a_{i+1,\, i+1}$, $a_{i+1,\, k}$ and $a_{k,\, k}$ are subjected to the row and column operations involving $\cos\theta_{i+1,k}$ and $\sin\theta_{i+1,k}$. For $\ell = k + 1, k + 2, \ldots, n$ in turn, the part of the row transformation involving $\cos\theta_{i+1,\, k}$ and $\sin\theta_{i+1,\, k}$ is performed on $a_{i+1,\, \ell}$ and $a_{k,\, \ell}$. For $\ell = k + 1, k + 2, \ldots, n$ in turn, the part of the column transformation involving $\cos\theta_{i+1,\, \ell}$ and $\sin\theta_{i+1,\, \ell}$ is performed on $a_{i+1,\, k}$ and $a_{k,\, \ell}$. Now all the transformations involving the i^{th} major step have been performed on all the elements of row k and on elements i + 2, i + 3, ..., n of row i + 1.

5. The completed k^{th} row is transferred to auxiliary storage.

When substeps 4 and 5 have been completed for all appropriate values of k, the work on row i + 1 has also been completed. The values of $\cos\theta_{i+1,\ k}$ and $\sin\theta_{i+1,\ k}$ for $k = i + 2, i + 3, \ldots, n$ are transferred to auxiliary storage and row i + 1 is transferred to the group 1 core storage locations. Since the $i + 1^{st}$ row plays the same part in the next major step as did the i^{th} in the step just described, everything is ready for substep 2 in the next major step. Substep 1 is, in fact, only required in the first major step because the appropriate row is already in the storage locations of group 1 in subsequent steps.

10.2.3 Extraction of the Eigenvalues of a Tridiagonal Matrix

NASTRAN employs the Q-R transformation of Francis (Reference 5) which is an orthogonal transformation $[A_{r+1}] = [Q_r]^T [A_r] [Q_r]$ such that $[A_r]$ may be factored into the product $[Q_r] [R_r]$ where $[R_r]$ is an upper triangular matrix. Thus

$$[A_r] = [Q_r][R_r] \ , \tag{16}$$

and

$$\begin{aligned}[A_{r+1}] &= [Q_r]^T[A_r][Q_r] \\ &= [Q_r]^T[Q_r][R_r][Q_r] \ .\end{aligned}$$

Now $[Q_r]^T[Q_r] = [I]$ by virtue of the orthogonality property, Equation 3, so that

$$[A_{r+1}] = [R_r][Q_r] \ . \tag{17}$$

It follows that $[A_{r+1}]$ is determined from $[A_r]$ by performing in succession the decomposition given by Equation 16 and the multiplication given by Equation 17. Francis has shown that if a matrix $[A] = [A_1]$ is nonsingular, then, in the limit as $r \to \infty$, $[A_r]$ will approach an upper triangular matrix; because eigenvalues are preserved under orthogonal transformation, it follows that diagonal elements of the limiting matrix are the eigenvalues of the original matrix [A]. Although the method can be applied to any matrix, it is particularly suitable for tridiagonal matrices because the bandwidth of the matrix can be preserved, as will be shown. In the case where [A] is symmetric, the matrix $[A_r]$ will, of course, tend to a diagonal form as $r \to \infty$.

It may be shown that the upper triangular matrix, $[R_r]$, and the orthogonal matrix $[Q_r]$ are unique, Reference 4, page 241; several methods, however, are available for performing the decomposition. In the method of calculation devised by Francis, $[Q_r]$ is expressed as a product of (n-1) elementary rotation matrices, where n is the order of $[A_r]$:

$$[Q_r] = [T^{(1)}][T^{(2)}] \cdots [T^{(n-1)}] \, . \tag{18}$$

The nonzero elements of the jth elementary rotation matrix are

$$\begin{aligned} t^{(j)}_{j,\,j} &= t^{(j)}_{j+1,\,j+1} = c_j \, , \\ t^{(j)}_{j+1,\,j} &= -t^{(j)}_{j,\,j+1} = s_j \, , \\ t^{(j)}_{k,\,k} &= 1, \; k \neq j \text{ or } j+1. \end{aligned} \tag{19}$$

The manner in which the c_j and s_j coefficients are obtained from the elements of $[A_r]$ will be shown shortly. $[R_r]$ is, from Equation 16 and the orthogonality property,

$$\begin{aligned} [R_r] &= [Q_r]^{-1}[A_r] = [Q_r]^T[A_r] \\ &= [T^{(n-1)}]^T[T^{(n-2)}]^T \cdots [T^{(2)}]^T[T^{(1)}]^T[A_r] \, . \end{aligned} \tag{20}$$

Let the nonzero elements of $[A_r]$, $[A_{r+1}]$ and $[R_r]$ be defined as follows:

$$[A_r] = \begin{bmatrix} a_1 & b_2 & & & 0 \\ b_2 & a_2 & b_3 & & \\ & \cdot & \cdot & \cdot & \\ & & b_{n-1} & a_{n-1} & b_n \\ 0 & & & b_n & a_n \end{bmatrix} \tag{21}$$

$$[A_{r+1}] = \begin{bmatrix} \bar{a}_1 & \bar{b}_2 & & & 0 \\ \bar{b}_2 & \bar{a}_2 & \bar{b}_3 & & \\ & \cdot & \cdot & \cdot & \\ & & \bar{b}_{n-1} & \bar{a}_{n-1} & \bar{b}_n \\ 0 & & & \bar{b}_n & \bar{a}_n \end{bmatrix} , \tag{22}$$

and

$$[R_r] = \begin{bmatrix} r_1 & q_1 & t_1 & & 0 \\ & r_2 & q_2 & t_2 & \\ & & \cdot & \cdot & \cdot \\ & 0 & & r_{n-1} & q_{n-1} \\ & & & & r_n \end{bmatrix} . \tag{23}$$

The coefficients of the elementary rotation matrices are selected so as to reduce the subdiagonal terms of $[R_r]$ to zero. Specifically,

$$\left.\begin{aligned} s_j &= \frac{b_{j+1}}{\left(p_j^2 + b_{j+1}^2\right)^{\frac{1}{2}}} , \\ c_j &= \frac{p_j}{\left(p_j^2 + b_{j+1}^2\right)^{\frac{1}{2}}} , \end{aligned}\right\} \quad j = 1, 2, \ldots, n-1 , \tag{24}$$

where

$$\left.\begin{aligned} p_1 &= a_1 , \\ p_2 &= c_1 a_2 - s_1 b_2 , \\ p_j &= c_{j-1}\, a_j - s_{j-1}\, c_{j-2}\, b_j . \qquad j = 3, 4, \ldots, n-1 . \end{aligned}\right\} \tag{25}$$

Substitution into Equation 20 yields the elements of $[R_r]$.

$$\left.\begin{aligned}
r_j &= c_j p_j + s_j b_{j+1} \, , && j = 1, 2, \ldots, n-1 \, , \\
r_n &= p_n \, , && \\
q_1 &= c_1 \, b_2 + s_1 \, a_2 \, , && \\
q_j &= s_j \, a_{j+1} + c_j \, c_{j-1} \, b_{j+1} \, , && j = 2, 3, \ldots, n-1 \, , \\
t_j &= s_j \, b_{j+2} \, , && j = 1, 2, \ldots, n-2 \, .
\end{aligned}\right\} \qquad (26)$$

The elements of $[A_{r+1}]$ are, from Equation 17,

$$\left.\begin{aligned}
\bar{a}_1 &= c_1 \, r_1 + s_1 \, q_1 \, , && \\
\bar{a}_j &= c_{j-1} \, c_j \, r_j + s_j \, q_j \, , && j = 2, 3, \ldots, n-1 \, , \\
\bar{a}_n &= c_{n-1} \, r_n \, , && \\
\bar{b}_{j+1} &= s_j \, r_{j+1} \, , && j = 1, 2, \ldots, n-1 \, .
\end{aligned}\right\} \qquad (27)$$

NASTRAN uses a variation, Reference 6, of Francis' original method that avoids the calculation of square roots. This is done by using the following equations in place of Equation 27

$$\left.\begin{aligned}
\bar{a}_j &= (1 + s_j^2) \, g_j + s_j^2 \, a_{j+1} \, , && j = 1, 2, \ldots, n-1 \, ,
\end{aligned}\right\} \qquad (28)$$

$$\left.\begin{aligned}
\bar{a}_n &= g_n \, , && \\
\bar{b}_{j+1}^2 &= s_j^2 \, (p_{j+1}^2 + b_{j+2}^2) \, , && j = 1, 2, \ldots, n-2 \, , \\
\bar{b}_n^2 &= s_{n-1}^2 \, p_n^2 \, , &&
\end{aligned}\right\} \qquad (29)$$

Where,

$$\left.\begin{aligned} g_1 &= a_1 \,, \\ g_j &= c_{j-1}\, p_j = a_j - s_{j-1}^2 (a_j + g_{j-1}) \,, \qquad j = 2, 3, \ldots, n \,, \\ p_1^2 &= a_1^2 \,, \\ p_j^2 &= g_j^2 / c_{j-1}^2 \qquad \text{if} \quad c_{j-1} \neq 0 \,, \qquad j = 2, 3, \ldots, n \,, \\ &= c_{j-2}^2\, b_j^2 \qquad \text{if} \quad c_{j-1} = 0 \,, \qquad j = 2, 3, \ldots, n \,. \end{aligned}\right\} \tag{30}$$

The reason that the use of Equations 28, 29 and 30 in place of Equation 27 avoids the calculation of square roots can best be seen by considering the terms input to and produced by these equations.

For Equation 27, the input terms consist of the elements of $[A_r]$. That is; a_1, a_2, ..., a_n and b_2, b_3, ..., b_n. The data produced is $\bar{a}_1$, $\bar{a}_2$, ..., $\bar{a}_n$ and $\bar{b}_2$, $\bar{b}_3$, ..., $\bar{b}_n$ which are the elements of $[A_{r+1}]$. Hence, this completes one iteration step, but involves the calculation of square roots.

However, for Equations 28, 29, and 30, the input terms consist of a_1, a_2, ..., a_n and b_1^2, b_2^2, ..., b_n^2. The data produced are $\bar{a}_1$, $\bar{a}_2$, ..., $\bar{a}_n$ and $\bar{b}_1^2$, $\bar{b}_2^2$, ..., $\bar{b}_n^2$ which serve as the input to the next iteration step. Here no square roots need to be computed.

Convergence of the tridiagonal matrix to a diagonal form is speeded up by origin shifting. NASTRAN employs a procedure suggested by Wilkinson (Reference 7), who has shown that when the eigenvalues are real, the best shift strategy is to subtract a_n from each diagonal element of [A], thereby reducing each eigenvalue by a_n.

Another useful device in speeding up the determination of eigenvalues of tridiagonal matrices takes advantage of zeroes which may occur in the off-diagonal rows. Let the $[A_r]$ matrix be represented in partitioned form as follows.

$$\left[\begin{array}{c|c|c}
\begin{matrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & & \ddots & & \\ & & & a_{j-2} & b_{j-2} \\ & & & b_{j-2} & a_{j-1} \end{matrix} & & \\ \hline
& \begin{matrix} a_j & b_j & & & \\ b_j & a_{j+1} & b_{j+1} & & \\ & & \ddots & & \\ & & b_{m-2} & a_{m-1} & b_{m-1} \\ & & & b_{m-1} & a_m \end{matrix} & \\ \hline
& & \begin{matrix} a_{m+1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & a_n \end{matrix}
\end{array}\right] \tag{31}$$

In this matrix the diagonal terms in the lower right hand partition are eigenvalues that have been determined in previous iterations. The j^{th} row is the next lowest in which the off-diagonal term, b_{j-1}, vanishes, thereby uncoupling the equations in the first j-1 rows from the rest. As a result, the eigenvalues of the matrix in the central block may be obtained separately. Other uncoupled blocks may be found in the upper left partition.

The iteration described by Equations 28, 29 and 30 is continued until $\bar{b}_{m-1}^2$ vanishes to satisfactory accuracy so that $\bar{a}_m$ may be accepted as an eigenvalue of the shifted matrix. $\bar{b}_{m-1}^2$ must be negligible compared to $\bar{a}_m^2$. ($\bar{b}_{m-1}^2 + \bar{a}_m^2$ must approximately equal $\bar{a}_m^2$). $\bar{a}_m$ is transferred to the lower partition and the process is continued until all the eigenvalues of the partitioned matrix have been extracted.

10.2.4 Computation of Eigenvectors

The eigenvector corresponding to any eigenvalue, λ_i, of the tridiagonal matrix may be determined by solving (n-1) of the equations:

$$\left.\begin{aligned}
(a_1 - \lambda_i)\, x_1 + b_2 x_2 &= 0 \quad , \\
b_2\, x_1 + (a_2 - \lambda_i)\, x_2 + b_3 x_3 &= 0 \quad , \\
&\ \ \vdots \\
b_{n-1}\, x_{n-1} + (a_{n-1} - \lambda_i)\, x_{n-1} + b_n x_n &= 0 \quad , \\
b_n\, x_{n-1} + (a_n - \lambda_i)\, x_n &= 0 \quad .
\end{aligned}\right\} \qquad (32)$$

If the first (n - 1) equations are used, the solution is obtained by taking x_1 to be unity and back substituting in the equations to obtain values of $x_2, x_3, \ldots, x_n$. Wilkinson (Reference 7) has shown that this method is unstable and has suggested the following approach which is used in NASTRAN.

The tridiagonal matrix $[A - \lambda_i\, I]$ is factored into the product of a lower unit triangle $[L_i]$ and an upper triangle $[U_i]$. In the decomposition partial pivoting is used, i.e., the pivotal row at each stage is selected to be that equation which has the largest coefficient of the variable being eliminated. At each stage there will be only two equations containing that variable. The eigenvector $\{\phi_i\}$ is then obtained from solution of the equation

$$[U_i]\{\phi_i\} = \{C\} \quad , \qquad (33)$$

where {C} is arbitrarily selected. The solution is easily obtained by back substitution because $[U_i]$ has the form

$$[U_i] = \begin{bmatrix} p_1 & q_1 & r_1 & & & \\ & p_2 & q_2 & r_2 & & \\ & & & \ddots & & \\ & & & p_{n-2} & q_{n-2} & r_{n-2} \\ & & & & p_{n-1} & q_{n-1} \\ & & & & & p_n \end{bmatrix} . \qquad (34)$$

An improved solution is obtained by repeated application of Equation 33 using the current estimate of $\{\phi_i\}$ on the right hand side. Thus

$$[U_i]\{\phi_i^{(n)}\} = \{\phi_i^{(n-1)}\} \ , \tag{35}$$

where $\{\phi_i^{(0)}\} = \{C\}$. Wilkinson, Reference 7, shows that, if the computed eigenvalue λ_i is a close approximation to the true eigenvalue, convergence is so rapid that more than two iterations are never required. The test applied is that the maximum component of the eigenvector does not change (to single precision accuracy) in one iteration. The initial vector $\{C\}$ is chosen so that each element is unity.

In the case of a double eigenvalue, the above method gives one eigenvector $\{\phi_1\}$. If we start with any initial vector $\{b\}$ orthogonal to $\{\phi_1\}$ and apply the previous algorithm, convergence to $\{\phi_2\}$, the other eigenvector, will result. The following procedure is used in NASTRAN to extract multiple eigenvalues. If eigenvectors $\{\phi_1\}, \{\phi_2\}, \ldots, \{\phi_m\}$ with elements $\{\phi_s\} = \lfloor a_{1s}, a_{2s}, \ldots, a_{ns} \rfloor^T$ have been obtained, an initial vector $\{b\}$ orthogonal to each of these eigenvectors is obtained by taking the components $b_{m+1}, b_{m+2}, \ldots, b_n$ as unity and solving the simultaneous equations

$$\left.\begin{aligned}
b_1\, a_{11} + b_2\, a_{21} + \ldots + b_m\, a_{m1} &= -\sum_{s=m+1}^{n} (a_{s1}) \ , \\
b_1\, a_{12} + b_2\, a_{22} + \ldots + b_m\, a_{m2} &= -\sum_{s=m+1}^{n} (a_{s2}) \ , \\
&\ \ \vdots \\
b_1\, a_{1m} + b_{2m}\, a_{2m} + \ldots + b_m a_{mm} &= -\sum_{s=m+1}^{n} (a_{sm}) \ .
\end{aligned}\right\} \tag{36}$$

Accumulated rounding errors will result in the computed multiple eigenvectors not being exactly orthogonal to one another. The following Gram-Schmidt algorithm described by Bodewig (Reference 8) is used to produce an orthogonal set of k eigenvectors $\{y_s\}$ from the almost orthogonal set $\{x_s\}$. For s = 1, select

$$\{y_1\} = \{x_1\} \;/\; ||\ \{x_1\}\ || \ . \tag{37}$$

Then for $1 < s \leq k$, calculate

$$\left.\begin{aligned} \{z_s\} &= \{x_s\} - \sum_{t=1}^{s-1} (\{x_s\}^T \{y_t\}) \{y_t\} \text{ , and} \\ \{y_s\} &= \{z_s\} / \| \{z_s\} \| \text{ ,} \end{aligned}\right\} \qquad (38)$$

where $\| \{z_s\} \|$ denoted the Euclidean norm, $\sqrt{z_{s1}^2 + z_{s2}^2 + \dots + z_{sn}^2}$, of a vector $\lfloor z_{s1}, z_{s2}, \dots, z_{sn} \rfloor^T$, and where $\{x_s\}^T \{y_t\}$ is a scalar product of the vectors $\{x_s\}$ and $\{y_t\}$.

Finally, when all the eigenvectors of the tridiagonal matrix have been extracted, the matrix multiplications of Equation 8 are carried out to obtain the eigenvectors of the original matrix [A].

EIGENVALUE EXTRACTION METHODS

REFERENCES

1. Givens, W., "Numerical Computation of the Characteristic Values of a Real Symmetric Matrix," Oak Ridge National Lab., ORNL-1574, 1954.

2. Lanczos, C., "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators," J. Res. Nat. Bur. Stand., Vol. 45, 1950, pp. 255-282.

3. Householder, A. S., and Bauer, F. L., "On Certain Methods for Expanding the Characteristic Polynomial," Numerische Methematik, Vol. 1, 1959, pp. 29-37.

4. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965, pp. 506-510.

5. Francis, J. G. F., "The QR Transformation, a Unitary Analogue to the LR Transformation," The Computer Journal, Vol. 4, No. 3 (Oct. 1961) and No. 4 (Jan. 1962).

6. Ortega, J. M. and Kaiser, H. F., "The LL^T and QR Methods for Symmetric Tridiagonal Matrices," The Computer Journal, Vol. 6, No. 1, January 1963, pp. 99-101.

7. Wilkinson, J. H., "The Calculation of the Eigenvectors of Codiagonal Matrices," The Computer Journal, Vol. I, 1958, p. 90.

8. Bodewig, E., Matrix Calculus, Interscience Publishers, 1959.

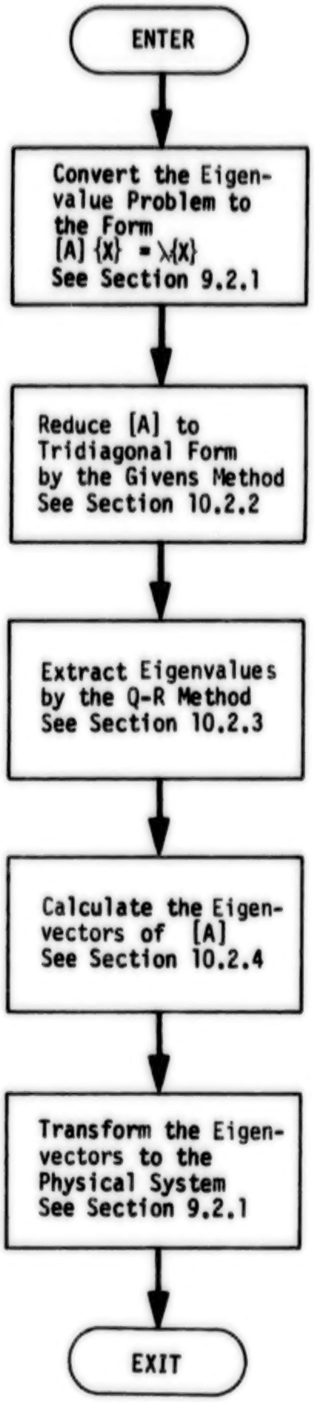

ENTER
Convert the Eigenvalue Problem to the Form
[A] {X} = λ{X}
See Section 9.2.1
Reduce [A] to Tridiagonal Form by the Givens Method
See Section 10.2.2
Extract Eigenvalues by the Q-R Method
See Section 10.2.3
Calculate the Eigenvectors of [A]
See Section 10.2.4
Transform the Eigenvectors to the Physical System
See Section 9.2.1
EXIT

Figure 1. Simplified flow diagram for the tridiagonal method.

10.3 THE DETERMINANT METHOD

10.3.1 Fundamentals of the Determinant Method

The basic notion employed in the determinant method of eigenvalue extraction is very simple. If the elements in a matrix [A] are polynomial functions of the operator p, then the determinant of [A] can be expressed as

$$D(A) = (p - p_1)(p - p_2)(p - p_3) \ldots (p - p_n) \quad , \tag{1}$$

where $p_1, p_2, p_3 \ldots p_n$ are the eigenvalues of the matrix. The value of the determinant vanishes for $p = p_i$, $i = 1,2,3 \ldots n$.

In the determinant method, the determinant is evaluated for trial values of p, selected according to some iterative procedure, and a criterion is established to determine when D(A) is sufficiently small or when p is sufficiently close to an eigenvalue. The eigenvector is then found by solution of the equation

$$[A]\{u\} = 0 \quad , \tag{2}$$

with one of the elements of {u} preset.

The most convenient procedure for evaluating the determinant of a matrix employs the triangular decomposition

$$[A] = [L][U] \quad , \tag{3}$$

where [L] is a lower unit triangular matrix (unit values on the diagonal) and [U] is an upper triangular matrix. The determinant of [A] is equal to the product of the diagonal terms of [U].

Two versions of the triangular decomposition operation are provided. In the standard version row interchanges are used in order to improve numerical stability. In an optional version, which is available for real eigenvalue extraction only, row interchanges are not used. The optional version is approximately four times as fast as the standard version for banded matrices, but it includes a small risk of numerical failure due to the fact that [A] is seldom a positive definite matrix. The algorithms employed in both versions are described in Section 2.2.

The matrix [A] may be expressed as

$$[A] = -p[M] + [K] \quad , \tag{4}$$

PAGE 641 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

642

for real eigenvalue problems, and as

$$[A] = p^2[M] + p[B] + [K] \quad , \tag{5}$$

for complex eigenvalue problems.

The determinant method is not a particularly efficient method because a large number of triangular decompositions of the [A] matrix are performed if more than a few eigenvalues are desired. An exception occurs in the case of very narrowly banded matrices, such as those in beam problems, where matrix decomposition is a relatively rapid operation. The main strength of the determinant method is its insensitivity to the functional form of the elements of the [A] matrix which could, for example, contain poles as well as zeroes, or be transcendental functions of p. These effects occur frequently in hydroelastic and aeroelastic problems.

10.3.2 Iteration Algorithm

Wilkinson's recent, but already standard, treatise (Reference 1) includes an authoritative discussion of polynomial curve-fitting schemes for tracking the roots of a determinant. He shows that little is to be gained by using polynomials higher than the second degree. Accordingly, Muller's quadratic method (Reference 1, page 435) is used in NASTRAN. The form of the algorithm is as follows.

Consider a series of determinants, D_{k-2}, D_{k-1}, D_k, evaluated for trial values of the eigenvalue, $p = p_{k-2}, p_{k-1}, p_k$. A better approximation to the eigenvalue is obtained from the following calculations. Let

$$h_k = p_k - p_{k-1} \quad , \tag{6}$$

$$\lambda_k = \frac{h_k}{h_{k-1}} \quad , \tag{7}$$

$$\delta_k = 1 + \lambda_k \quad . \tag{8}$$

Then

$$h_{k+1} = \lambda_{k+1} h_k \quad , \tag{9}$$

$$p_{k+1} = p_k + h_{k+1} \quad , \tag{10}$$

where

$$\lambda_{k+1} = \frac{-2D_k\delta_k}{g_k \pm \left[g_k^2 - 4D_k\delta_k\lambda_k\left(D_{k-2}\lambda_k - D_{k-1}\delta_k + D_k\right)\right]^{1/2}} \quad , \tag{11}$$

in which

$$g_k = D_{k-2}\lambda_k^2 - D_{k-1}\delta_k^2 + D_k(\lambda_k + \delta_k) \quad . \tag{12}$$

The (+) or (-) sign in Equation (11) is selected to minimize the absolute value of λ_{k+1}. In the case when p_k, p_{k-1}, and p_{k-2} are all arbitrarily selected initial values (starting points), the starting points are arranged such that

$$D_k \leqslant D_{k-1} \leqslant D_{k-2} \quad , \tag{13}$$

and the (+) or (-) sign in Equation 11 is selected to minimize the distance from p_{k+1} to the closest starting point rather than to p_k.

In a real eigenvalue analysis, it is possible to calculate a complex value of λ_{k+1} from Equation 11. In order to preclude the occurrence of complex arithmetic in a real eigenvalue analysis, only the real part of λ_{k+1} is used to estimate p_{k+1}. The real part corresponds to the minimum absolute value of the parabolic approximation.

10.3.3 Scaling

In calculating the determinant of [A] (Equation 3), some form of scaling must be employed, because the accumulated product will rapidly overflow or underflow the floating point size of a digital computer. Accordingly, the accumulated product of the diagonal terms of [U] is calculated and stored as a scaled number

$$D = d \times 10^n \quad , \tag{14}$$

where

$$\frac{1}{10} \leq d < 1 \quad . \tag{15}$$

The arithmetic operations indicated in Equations 11 and 12 are calculated in scaled arithmetic. λ_{k+1} is then reverted to unscaled form.

ORIGINAL PAGE IS OF POOR QUALITY

10.3.4 Sweeping of Previously Extracted Eigenvalues

Once an eigenvalue has been found to satisfactory accuracy, a return to that eigenvalue by the iteration algorithm can be prevented by dividing the determinant by the factor $(p - p_i')$, where p_i' is the accepted approximation to p_i, in all subsequent calculations.

Thus

$$D^{(1)}(A) = \frac{D(A)}{p - p_1'} \quad , \tag{16}$$

should be used in place of D(A) after the first eigenvalue has been found. In general, the reduced determinant used for finding the i+1st eigenvalue is

$$D^{(i)}(A) = \frac{D^{(i-1)}(A)}{p - p_i'} = \frac{D(A)}{(p - p_1')(p - p_2') \; . \; . \; . \; (p - p_i')} \quad . \tag{17}$$

Wilkinson states that this sweeping procedure is quite satisfactory provided that all p_i' have been calculated to an accuracy that is limited only by round-off error.

In some instances there are known eigenvalues for which calculations need not be made. An important example is the set of rigid body (zero-frequency) modes that are calculated by a separate subroutine (see Section 9.2). In addition, the user may know of other eigenvalues, such as those extracted in a previous execution or those resulting from transfer functions (see Section 9.3), that should be avoided. A special data card (EIGP) is used in complex analysis to specify the location of such roots. They are then eliminated from the determinant by a preliminary operation

$$D^{(0)}(A) = \frac{D(A)}{\prod_k (p - p_k)^{m_k}} \quad , \tag{18}$$

where m_k is the multiplicity of the known eigenvalue, p_k.

For problems with conjugate complex eigenvalues (complex eigenvalue analysis with real matrices) the conjugates of the extracted eigenvalues are also swept from the determinant. Thus

$$D^{(i)}(A) = \frac{D^{(i-1)}(A)}{(p - p_i')(p - \bar{p}_i')} \quad , \tag{19}$$

where $\bar{p}_i'$ is the conjugate of p_i'.

There is a danger in this procedure that if p_i' is very near the axis of reals, the exact eigenvalue may, in fact, be real. To avoid this situation a test is applied to the imaginary part of p_i' and the conjugate of p_i' is swept only if

$$|\text{Im } p_i'| \geq 1000.0(R_{max})(\varepsilon) \quad . \tag{20}$$

The parameters R_{max} and ε are discussed in the next two sections.

10.3.5 Search Procedures

Three initial values of p (starting points) are needed in order to start the iteration algorithm. The determinant method is essentially a root-tracking method that finds nearby roots with ease and remote roots with difficulty. Thus it is not advisable to use the same three starting points for all eigenvalues because the eigenvalues are usually distributed throughout a region of the p-plane and at lease some of the eigenvalues will be remote from any given set of three starting points.

In the case of real eigenvalue analysis, the starting points are initially uniformly distributed in an interval of p. The user specifies the lowest and highest eigenvalues of interest, R_{min} and R_{max}, and also supplies an estimate, N_E, of the number of roots in the range. A total of $2N_E + 1$ starting points are then initially located such that the first coincides with R_{min} and the rest are placed at uniform intervals between R_{min} and R_{max}. It has, however, been found that this distribution of starting points, while satisfactory in most cases, causes computational problems when the higher frequency in any interval is larger than the lower frequency in that interval by several orders of magnitude. In order to avoid this difficulty, any initial interval in which the higher frequency is larger than (or equal to) ten times the lower frequency in that range is further subdivided into smaller unequal intervals, resulting in additional unequally spaced (as per a geometrical progression) starting points. This scheme thus guarantees that every starting point has a frequency that is less than ten times the frequency of the next lower starting point. Accordingly, the resulting number of starting points may therefore be larger than $2N_E + 1$.

The process of extracting eigenvalues is initiated from the three smallest starting points, p_{s1}, p_{s2}, and p_{s3}. After all eigenvalues within the "range" of these starting points have been extracted, p_{s1} is dropped, p_{s4} is added, and the search is repeated. The details will be described later.

The search procedure for complex eigenvalues is more complicated because the roots are distributed throughout a plane rather than along a line. Fortunately, in structural analysis, most roots are found near the imaginary axis so that a set of starting points placed along the imaginary axis will usually produce satisfactory results. In order to provide for more general problems, rectangular search regions may be located by the user in any part of the complex plane as shown in Figure 1. It is intended that all eigenvalues within the search regions be extracted (as limited by the desired maximum number of roots). There may be any number of search regions, and the search regions may overlap. The search regions are established by means of the coordinates of end points (A_j, B_j) and the width of the region (ℓ_j). The user estimates the number of roots

(N_{ej}) in each search region. For problems with real coefficients, all roots are either real or they occur in conjugate complex pairs, and it is inefficient for the user to specify regions in the lower half of the complex plane since the existence of conjugate pairs is assumed. The user will not, however, be prevented from specifying a search in the lower half of the complex plane since it is useful, in many cases, for a search region to include the origin or a band along the real axis.

The steps in the search procedure for complex eigenvalue extraction are as follows:

1. Select $2N_{ej} + 2$ starting points equally distributed along the line A_j, B_j. Do not use A_j and B_j as starting points.
2. Find the starting point in Region I that is nearest to the origin as shown in Figure 2 and designate it p_{s1}. Draw a line perpendicular to A_1B_1 midway between p_{s1} and p_{s1}', the point next nearest to the origin, thereby dividing Region I into Regions IA and IB.
3. Select the three starting points in Region IA nearest to the line, a - a', as the initial set and proceed to extract roots. Then proceed to the three starting points in Region IB (p_{s1}', p_{s2}', p_{s3}') that are closest to the line a - a'. Then return to the points p_{s2}, p_{s3}, p_{s4} in Region IA, and, in general, alternate back and forth between the two regions until all starting points have been used once, or until termination occurs for some other reason.
4. When all starting points in Region I have been used once, proceed to Region II, etc. Sweep out all previously extracted roots from whatever region, in evaluating determinants.
5. When all starting points in all regions have been used once, return to Region I, II, etc. in turn and repeat the above procedure. Continue to repeat until no new roots are found in any pass through the regions or until termination occurs for some other reason.

When searching for either real or complex roots from any set of starting points, the search is terminated if a root is predicted to lie outside of the local search region. In the case of complex analysis, the local search region for the current starting point set is bounded by lines parallel to A_jB_j at a distance $\frac{\ell}{2}$ and extends from the first starting point on the other side of a - a' to a line that is 55% of the way between the current starting point that is farthest from a - a' and the starting point that will be picked up in the next set (see Figure 2).

PAGE 10.3-5B INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

In the case of real eigenvalue analysis the local search region extends from R_{min} to a point that is 55% of the way from the largest current starting point to the one that will be picked up in the next set. The reason for permitting the local search region to extend all the way back to R_{min} is to improve the chance of finding eigenvalues that were initially bypassed. Additional assurance is obtained by making repeated passes through all of the starting points until no new eigenvalues are found on one complete pass. The computational effort for the last pass is trivial because the determinants for all starting points are saved.

The search for eigenvalues is finally terminated when no additional roots are found in one complete pass through all regions or when the maximum desired number of roots for each region, N_{dj}, as requested by the user, have been extracted.

Failure to find additional roots normally occurs because all of the roots within the desired region(s) have been extracted. Situations can occur in which some roots will be missed. The most common occurs in complex eigenvalue analysis when one or more desired roots lie at a large distance from the centerline of the region, see Figure 2, and several other roots lie just beyond the ends of the region. The search procedure is likely to be attracted toward the latter roots and may not find the former. The possibility of such failures is one reason for permitting multiple search regions. Another possible reason for missing roots is failure of the iteration algorithm to converge, as discussed below.

10.3.6 Convergence Criteria

The convergence criteria are based on successive values of the increment, h_k, in the estimated eigenvalue. No tests on the magnitude of the determinant or on any of the diagonal terms of the triangular decomposition are necessary or desirable.

Wilkinson (Reference 1, page 437) shows that, for h_k sufficiently small, the magnitude of h_k is approximately squared for each successive iteration when using Muller's method to find isolated roots. This is an extremely rapid rate of convergence. In a very few iterations, the "zone of indeterminacy" is reached within which h_k remains small, but exhibits random behavior due to round-off error.

Wilkinson states that, if it is desired to calculate the root to the greatest possible precision, the convergence criterion for accepting p_k as a root should be

$$|h_{k+1}| > |h_k| \qquad (21)$$

We accept his advice, tempered by practical considerations. The first of these is that Equation 21 may be satisfied during the first few iterations while the root tracking algorithm is picking up the "scent". Thus it must, in addition, be required that $|h_k|$, $|h_{k-1}|$ and $|h_{k-2}|$ be reasonably small. The second practical consideration is that several iterations may be wasted within the zone of indeterminacy while waiting for Equation 21 to be satisfied. This can be avoided by accepting p_k if $|h_k|$ is sufficiently small. Finally, if the number of iterations becomes excessively large without satisfying a convergence criterion, it is best to give up and proceed to a new set of starting points.

Figure 3 is a flow diagram of a set of tests which meet the requirements discussed above for real eigenvalue problems. The tests are based on calculated values of $\overline{H}_1$, $\overline{H}_2$, and $\overline{H}_3$ which are defined as

$$\overline{H}_1 = \frac{|h_{k-1}|}{\sqrt{|p_k|}} \quad , \qquad (22)$$

$$\overline{H}_2 = \frac{|h_k|}{\sqrt{|p_k|}} \quad , \qquad (23)$$

$$\overline{H}_3 = \frac{|h_{k+1}|}{\sqrt{|p_k|}} \quad . \qquad (24)$$

where p_k = kth estimate of an eigenvalue, and $h_k = p_k - p_{k-1}$.

The standard test in Figure 3 for accepting an eigenvalue is

$$\overline{H}_3 \leqslant 2\varepsilon \sqrt{R_{max}} \quad . \qquad (25)$$

ε may be specified by the user. An equivalent expression, in terms of vibration mode frequencies, is

$$\frac{\omega_{k+1} - \omega_k}{\omega_{max}} \leqslant \varepsilon \quad , \qquad (26)$$

where ω_{max} is the highest frequency of interest. A similar set of tests are performed for

651

complex eigenvalue problems.

The magnitude of the convergence criterion, ε, should be selected as a compromise between running time and accuracy. If, for example, it is found that the algorithm fails to find roots because of an excessive number of iterations, the value of ε should be increased. If failure occurs because the number of iterations exceeds the iteration limit, N_{IT}, for two successive sets of starting points, the value of ε is increased by a factor of ten. If successive pairs of failures still occur, ε is again increased by a factor of ten, until the number of permissible changes in ε, N_ε, is exceeded. The user is informed of the reduced precision of the calculations.

The values of the convergence parameters used in NASTRAN are

$$\varepsilon = 10^{-11} \quad ,$$

$$N_{IT} = 20 \quad ,$$

$$N_\varepsilon = 4 \quad .$$

The above value of ε may be overridden by the user.

The convergence rates for the Muller method are quite satisfactory for isolated roots and for double roots. For higher order multiple roots, the asymptotic rate of convergence is linear, i.e.,

$$h_{k+1} = a \cdot h_k \quad , \quad a < 1 \quad . \tag{27}$$

Convergence is progressively slower as the multiplicity increases. Zero frequency roots are the only kind that commonly occur with high multiplicity in structural analysis and provision is made for eliminating them, or other known roots, beforehand (see Section 10.3.4).

Experience with the determinant method indicates that, on average, convergence to isolated roots is achieved in about six iterations. The number of iterations increases for problems with multiple roots, many close roots, or roots clustered just beyond the ends of the desired region.

10.3.7 Test for Closeness to Starting Point

Once an eigenvalue has been extracted it is tested for closeness to one of the starting points, and if it is found to be too close, the starting point is shifted. The reason for the shift is that the value of the determinant near a root is small and contains considerable round-off error, so that the value of the swept determinant, $D/(p - p_k)$ may have considerable error. The criterion for shifting in real eigenvalue analysis is

$$p_{s,i} - p_k < 400\varepsilon \sqrt{R_{max} p_k} \quad . \tag{28}$$

The magnitude of the shift is

$$(p_{s,i})_{shifted} = p_{s,i} + 2000\varepsilon \sqrt{R_{max} p_k} \quad . \tag{29}$$

10.3.8 Recovery of Eigenvectors

Once an approximate eigenvalue, p_j, has been accepted, the eigenvector is determined by back substitution into the previously computed triangular decomposition of $[A(p_j)]$. Now since

$$[A(p_j)]\ \{u\} = [L(p_j)][U(p_j)]\{u\} = 0 \quad , \tag{30}$$

and since $[L(p_j)]$ is nonsingular, only $[U(p_j)]$ is used. The last diagonal term in $[U(p_j)]$ is normally the only term with very small value. The normal appearance of $[U(p_j)]$ is as follows, for $n = 7$.

$$\begin{bmatrix} X & X & X & X & 0 & 0 & 0 \\ & X & X & X & X & 0 & 0 \\ & & X & X & X & X & 0 \\ & & & X & X & X & X \\ & 0 & & & X & X & X \\ & & & & & X & X \\ & & & & & & \delta \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{Bmatrix} = 0 \quad . \tag{31}$$

The terms in the upper right corner are zero due to bandwidth. δ is a very small number. The eigenvector may be extracted by assigning an arbitrary value (such as 1.0) to u_7 and solving successfully for u_6, u_5, etc., from the preceding rows. Note that this is equivalent to placing a load vector $\{F\}$ on the right hand side that is null except for the last term which is set equal to δ.

Situations may occur in which U_{nn} is not the smallest diagonal term. Let U_{ii} be the smallest diagonal term with $i < n$. The most common reason for this occurrence is that the degrees of freedom u_{i+1}, u_{i+2} . . . u_n are, for some reason, not coupled to the preceding degrees of freedom. In this case all of the elements in the ith row of $[U(p_j)]$ will be very small as shown below for $i = 4$, $n = 7$.

$$\begin{bmatrix} X & X & X & X & 0 & 0 & 0 \\ & X & X & X & X & 0 & 0 \\ & & X & X & X & X & 0 \\ & 0 & & \delta_{44} & \delta_{45} & \delta_{46} & \delta_{47} \\ & & & & X & X & X \\ & & & & & X & X \\ & & & & & & X \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{Bmatrix} = 0 \quad . \qquad (32)$$

In the event of multiple or pathologically close eigenvalues, two or more rows or $[U(p_j)]$ will consist of very small values, exhibited below for the very exceptional case when U_{nn} is not very small.

$$\begin{bmatrix} X & X & X & X & 0 & 0 & 0 \\ & X & X & X & X & 0 & 0 \\ & & X & X & X & X & 0 \\ & 0 & & \delta_{44} & \delta_{45} & \delta_{46} & \delta_{47} \\ & & & & X & X & X \\ & & & & & \delta_{66} & \delta_{67} \\ & & & & & & X \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{Bmatrix} = 0 \quad . \qquad (33)$$

In order to accommodate the exceptional cases described above with the more common case when only the last diagonal term is small, a full load vector, $\{F\}$, is used in the eigenvector calculations. A different load vector is formed for each eigenvalue to ensure that independent eigenvectors are calculated for multiple or pathologically close roots. The ith component of the vector $\{F\}$ is calculated by the following formula.

$$F_i = \frac{\delta(-1)^{ij}}{1 + \left(1 - \frac{i}{n}\right)j} \quad , \tag{34}$$

where

δ = smallest U_{ii} ,

j = eigenvalue count ,

n = number of rows .

There is a possibility that the smallest diagonal element of [U] may be exactly zero for some eigenvalue, i.e., that an accepted eigenvalue (p_j) may be an exact eigenvalue of the problem. When this situation occurs, the zero diagonal element is set equal to a small finite number.

Eigenvectors are checked for orthogonality and normalized (see Section 9.2).

655

THE DETERMINANT METHOD

REFERENCE

1. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

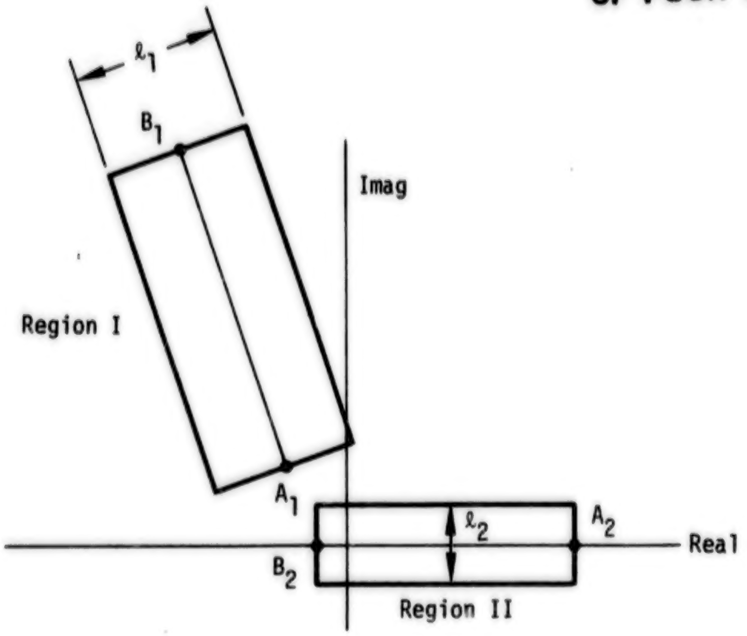

ℓ1
B1
Imag
Region I
A1
A2
ℓ2
Real
B2
Region II

Figure 1. Search regions.

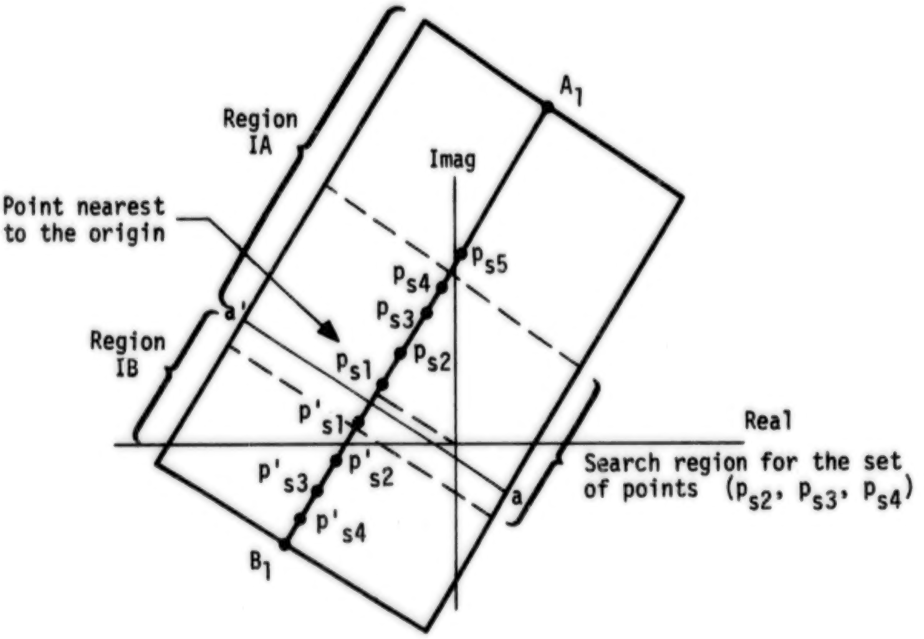

Region IA
A1
Imag
Point nearest to the origin
p_s5
p_s4
p_s3
p_s2
Region IB
a'
p_s1
p'_s1
Real
p'_s2
p'_s3
Search region for the set of points (p_s2, p_s3, p_s4)
a
p'_s4
B1

Figure 2. Location of starting points in a region.

ORIGINAL PAGE IS OF POOR QUALITY

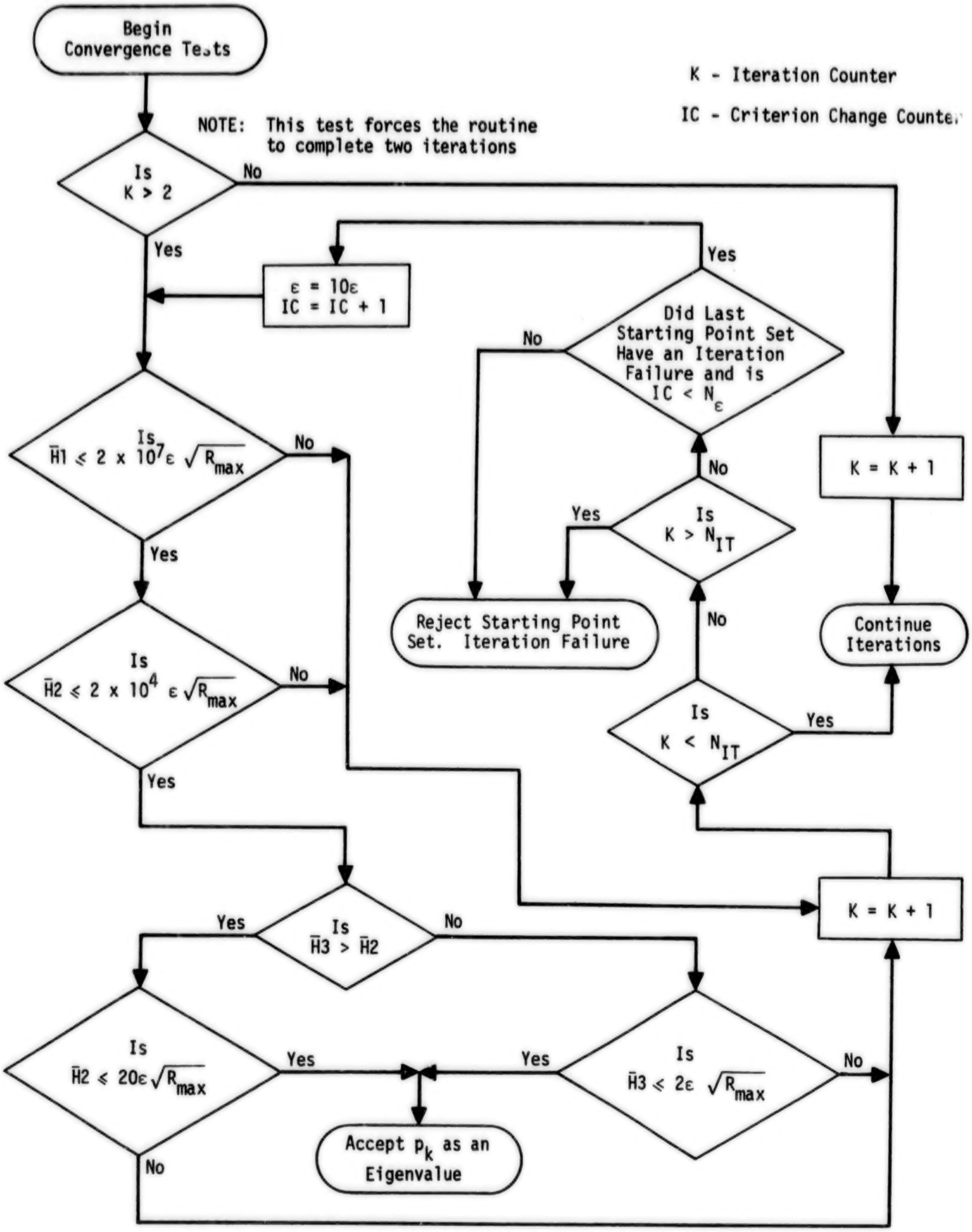

Begin Convergence Tests
K - Iteration Counter
IC - Criterion Change Counter
NOTE: This test forces the routine to complete two iterations
Is K > 2
No
Yes
ε = 10ε
IC = IC + 1
Did Last Starting Point Set Have an Iteration Failure and is IC < N_ε
Yes
No
Is $\bar{H}1 \leqslant 2 \times 10^7 \varepsilon \sqrt{R_{max}}$
No
Yes
K = K + 1
Is K > N_{IT}
Yes
No
Reject Starting Point Set. Iteration Failure
Continue Iterations
Is $\bar{H}2 \leqslant 2 \times 10^4 \varepsilon \sqrt{R_{max}}$
No
Yes
Is K < N_{IT}
Yes
No
K = K + 1
Is $\bar{H}3 > \bar{H}2$
Yes
No
Is $\bar{H}2 \leqslant 20\varepsilon \sqrt{R_{max}}$
Yes
No
Is $\bar{H}3 \leqslant 2\varepsilon \sqrt{R_{max}}$
Yes
No
Accept p_k as an Eigenvalue

Figure 3. Real eigenvalue convergence tests.

628

10.4 THE INVERSE POWER METHOD WITH SHIFTS

10.4.1 Introduction

The Inverse Power Method with Shifts is particularly effective for problems that are formulated by the displacement approach when only a fraction of all of the eigenvalues are required. The rudiments of the method are described in Wilkinson's recent book (Reference 1) where it is regarded chiefly as a powerful method for refining the accuracy of eigenvalues and eigenvectors that have been approximately located by other methods. In NASTRAN, the method is used as a stand-alone method to find all of the eigenvalues within a domain specified by the user.

It is well known that the standard inverse power method has a number of important defects for the solution of structural problems by the displacement approach. These include: awkwardness of procedure in the presence of zero eigenvalues (rigid body structural modes); slow convergence rates for closely spaced roots; and deterioration of accuracy in the higher modes as more roots are found. All of these defects are eliminated or minimized by the following modification of the method.

Let the eigenvalue problem be stated as follows,

$$[K - \lambda M]\{u\} = 0 \quad , \tag{1}$$

where, employing structural semantics, [K] is a stiffness matrix, [M] is a mass matrix, {u} is a displacement vector, and λ is the square of the natural frequency. Let

$$\lambda = \lambda_o + \Lambda \quad , \tag{2}$$

where λ_o is called the shift point. The iteration algorithm is

$$[K - \lambda_o M]\{w_n\} = [M]\{u_{n-1}\} \; , \tag{3}$$

$$\{u_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{4}$$

where C_n is equal to the value of the element of $\{w_n\}$ with largest absolute value. It is easy to prove that $1/C_n$ converges to Λ_1, the shifted eigenvalue nearest to the shift point, and that $\{u_n\}$

[illegible] 659 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

converges to the corresponding eigenvector of Equation 1. The following observations are pertinent.

1. A triangular decomposition of the matrix $[K - \lambda_0 M]$ is required in order to evaluate $\{w_n\}$ from Equation 3. The effort required to perform the triangular decomposition is greatly reduced if $[K - \lambda_0 M]$ is a narrow band matrix. Structural analysis by the displacement approach is characterized by narrowly banded stiffness and mass matrices.

2. It is unnecessary for [K] to be nonsingular, so that rigid body modes cause no special difficulty.

3. The shift point λ_0 may be changed (at the cost of an additional triangular decomposition) in order to improve the rate of convergence toward a particular eigenvalue, or to improve accuracy and convergence rate after several roots have been extracted from a given shift point.

4. λ_0 can be placed so as to obtain the eigenvalues within a desired frequency band and not just those that have the smallest absolute value.

The Inverse Power Method with Shifts can also be applied to complex eigenvalue problems. In NASTRAN the method is applied to problems stated in the quadratic form

$$[M\lambda^2 + B\lambda + K]\{u\} = 0 \quad , \qquad (5)$$

where [M], [B] and [K] may be real or complex, symmetrix or nonsymmetric, singular or nonsingular.

The development of the method is divided into separate subsections for real eigenvalue analysis, Section 10.4.2, and complex eigenvalue analysis, Section 10.4.4. In addition the procedures used by the program are summarized in Sections 10.4.3, real eigenvalue analysis, and Section 10.4.5, complex eigenvalue analysis.

ORIGINAL PAGE IS OF POOR QUALITY

10.4.2 Theory for Real Eigenvalue Analysis

10.4.2.1 Iteration Algorithm

The problem is to find the eigenvalues and eigenvectors for

$$[K - \lambda M]\{u\} = 0 \quad . \tag{1}$$

Both [K] and [M] are symmetric, but they may be singular. The eigenvalues may be either positive or negative, but we are predominantly interested in the positive eigenvalues. Eigenvalues may be multiple. It is required to find all the eigenvalues and eigenvectors within a specified range of λ.

Let

$$\lambda = \lambda_o + \Lambda \quad , \tag{2}$$

where λ_o is a constant called the shift point, so that Λ replaces λ as the eigenvalue parameter.

The algorithm used in the nth iteration step is

$$[K - \lambda_o M]\{w_n\} = [M]\{u_{n-1}\} \quad , \tag{3}$$

$$\{u_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{4}$$

where C_n is equal to the value of the element of $\{w_n\}$ with largest absolute value. It may be shown that $1/C_n$ converges to Λ_1, the shifted eigenvalue nearest to the shift point, and $\{u_n\}$ converges to the corresponding eigenvector. Convergence is proved as follows. In the proof use will be made of the orthogonality properties for the theoretically obtainable normalized exact eigenvectors, ϕ_i and ϕ_j, of Equation 1 which are

$$\phi_j^T M \phi_i = 0, \quad i \neq j \quad , \tag{5}$$

$$\phi_j^T M \phi_j = 1 \quad , \tag{6}$$

$$\phi_j^T K \phi_i = 0 \ , \ i \neq j, \text{ and} \tag{7}$$

$$\phi_j^T K \phi_j = \lambda_j \quad . \tag{8}$$

Note that the use of brackets to indicate matrix quantities has been omitted in the interest of conciseness. Expand the trial vectors, u_n and u_{n-1}, in terms of eigenvectors:

662

$$u_n = \sum_i \alpha_{i,n}\phi_i \quad , \tag{9}$$

$$u_{n-1} = \sum_i \alpha_{i,n-1}\phi_i \quad , \tag{10}$$

where $\alpha_{i,n}$ and $\alpha_{i,n-1}$ are coefficients. Substitute into Equation 3, premultiply by ϕ_j^T, and invoke the orthogonality conditions. Then it is seen that

$$(\lambda_j - \lambda_o)C_n\alpha_{j,n} = \alpha_{j,n-1} \quad , \tag{11}$$

or, substituting $\lambda_j = \lambda_o + \Lambda_j$,

$$P_{j,n} = \frac{\alpha_{j,n}}{\alpha_{j,n-1}} = \frac{1}{C_n(\lambda_j - \lambda_o)} = \frac{1}{C_n\Lambda_j} \quad . \tag{12}$$

Equation 12 shows that the relative proportion, $P_{j,n}$, of an eigenvector in successive trial vectors increases in inverse proportion to the magnitude of its shifted eigenvalue. Thus the proportion of the eigenvector that is closest to the shift point increases at the expense of the others and the process converges to the eigenvector whose eigenvalue is closest to the shift point. The product $C_n\Lambda_1$ converges to unity.

10.4.2.2 Convergence Criteria

The most common method for gaging the convergence of eigenvectors is to compare the difference of the elements of successive trial vectors, each of which has been normalized so that the largest element is unity, to an arbitrarily selected small number

$$u_n - u_{n-1} < \varepsilon \quad .$$

This method is not used in NASTRAN because it is unscientific in two respects:

1. The criterion ε is not related to the mass orthogonality test which is finally used to judge the suitability of eigenvectors.
2. Normalizing on the largest element of u_n neglects the fact that the sizes of elements are largely determined by scaling, (e.g., angles are usually much smaller than translations).

Closely related methods can be devised that overcome these objections and that also predict the rate of convergence. In NASTRAN, the convergence criteria are derived from the parameter ε that is supplied by the user for the mass orthogonality test

$$\bar{\phi}_i^T M \bar{\phi}_j < \varepsilon \qquad i \neq j \quad , \tag{13}$$

where the approximate eigenvectors, $\bar{\phi}_i$, have been normalized such that

$$\bar{\phi}_i^{\,T} M \bar{\phi}_i = 1 \quad . \tag{14}$$

The mass orthogonality test is applied after all the eigenvalues have been computed.

For a pair of well-converged approximate eigenvectors that have close eigenvalues but which are well removed from all other eigenvalues:

$$\bar{\phi}_1 \simeq \phi_1 + \varepsilon_{12}\phi_2 \quad , \tag{15}$$

$$\bar{\phi}_2 \simeq \phi_2 + \varepsilon_{21}\phi_1 \quad , \tag{16}$$

since it may be assumed on the basis of Equation 12 that the power method yields negligible contributions of all other eigenvectors. Thus

$$\bar{\phi}_1^{\,T} M \bar{\phi}_2 \simeq \varepsilon_{12} + \varepsilon_{21} \quad . \tag{17}$$

The program is able to estimate ε_{12} and ε_{21} from tests that will be described. A reasonable convergence criterion to be applied after each iteration is that

$$\varepsilon_{ij} < A\varepsilon \ , \tag{18}$$

for all i and j where ε is the criterion used in the mass-orthogonality test and A is a factor, less than unity, introduced to ensure passage of the mass-orthogonality test.

It has been shown (Equation 12) that the contributions of eigenvectors to the trial vector increase from iteration to iteration in inverse proportion to their shifted eigenvalues. Thus, during the late stages of iteration for ϕ_1, it may be assumed that only two eigenvectors have significant contributions:

$$u_n = \alpha_n(\phi_1 + \sqrt{\delta_n}\,\phi_2) \quad , \tag{19}$$

where $\sqrt{\delta_n} = \varepsilon_{12}$ is introduced for later convenience and α_{1n} (see Equation 9) is shortened to α_n. By virtue of Equation 12 the relative contribution of ϕ_2 in the preceding iteration step will be larger by the ratio of the shifted eigenvalues Λ_2/Λ_1. Thus for the two previous steps

$$u_{n-1} = \alpha_{n-1}\left(\phi_1 + \frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,\phi_2\right) , \tag{20}$$

$$u_{n-2} = \alpha_{n-2}\left(\phi_1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,\phi_2\right). \tag{21}$$

It will be assumed that $|\Lambda_2|>|\Lambda_1|$.

Applying the mass orthogonality properties, Equations 5 and 6, to Equations 19 and 20, it is seen that

$$u_n^T M u_n = \alpha_n^2(1+\delta_n) \quad , \tag{22}$$

$$u_{n-1}^T M u_{n-1} = \alpha_{n-1}^2\left(1+\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\delta_n\right) \quad . \tag{23}$$

Assuming $\left|\frac{\Lambda_2}{\Lambda_1}\right|\sqrt{\delta_n} \ll 1$ where $\left|\frac{\Lambda_2}{\Lambda_1}\right| > 1$, then to a very close approximation

$$\alpha_n \simeq \pm\left(u_n^T M u_n\right)^{1/2} \quad , \tag{24}$$

$$\alpha_{n-1} \simeq \pm\left(u_{n-1}^T M u_{n-1}\right)^{1/2} \quad . \tag{25}$$

From Equation 12, a good approximation to the eigenvalue is

$$\Lambda_1 = \frac{1}{C_n}\frac{\alpha_{n-1}}{\alpha_n} \simeq \frac{\pm 1}{C_n}\left(\frac{u_{n-1}^T M u_{n-1}}{u_n^T M u_n}\right)^{1/2} \quad , \tag{26}$$

where C_n is the normalizing factor introduced in Equation 4. The ambiguity in sign is removed by comparing the signs of selected components in u_n and u_{n-1}. A better approximation that can be evaluated when Λ_2/Λ_1 and δ_n are known is obtained by using Equations 22 and 23 for α_n and α_{n-1}.

$$\Lambda_1 = \frac{1}{C_n}\frac{\alpha_{n-1}}{\alpha_n} \simeq \pm\frac{1}{C_n}\left(\frac{u_{n-1}^T M u_{n-1}}{u_n^T M u_n}\right)^{1/2}\left(\frac{1+\delta_n}{1+\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\delta_n}\right)^{1/2} \quad . \tag{27}$$

Equation 27 shows that Equation 26 is good enough because δ_n must be extremely small in order that $\varepsilon_{12} = \sqrt{\delta_n}$ pass the eigenvector convergence test. Equation 26 is used in NASTRAN.

Turning now to the evaluation of δ_n and Λ_2/Λ_1, we first define the normalized difference of successive trial vectors,

$$\delta u_n = \frac{u_n}{\left(u_n^T M u_n\right)^{1/2}} - \frac{u_{n-1}}{\left(u_{n-1}^T M u_{n-1}\right)^{1/2}} \simeq \frac{u_n}{\alpha_n} - \frac{u_{n-1}}{\alpha_{n-1}} \quad . \tag{28}$$

Then, from Equations 19, 20 and 21:

$$\delta u_n = \left(1 - \frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\,\phi_2 \quad , \tag{29}$$

$$\delta u_{n-1} = \frac{\Lambda_2}{\Lambda_1}\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\;\phi_2 \quad , \tag{30}$$

and, by virtue of Equation 6

$$(\delta u_n)^T M \delta u_n = \left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n \quad , \tag{31}$$

$$(\delta u_n)^T M \delta u_{n-1} = \frac{\Lambda_2}{\Lambda_1}\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n \quad , \tag{32}$$

so that

$$\frac{\Lambda_2}{\Lambda_1} = \frac{(\delta u_n)^T M \delta u_{n-1}}{(\delta u_n)^T M \delta u_n} \quad , \tag{33}$$

and

$$\delta_n = \frac{(\delta u_n)^T M \delta u_n}{\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2} \quad . \tag{34}$$

The convergence criterion is

$$\sqrt{\delta_n} < A\varepsilon \quad . \tag{35}$$

Note that the application of the criterion requires three successive trial vectors. In some cases the rate of convergence may be so rapid that Λ_2/Λ_1, as computed by Equation 33, may have lost all numerical significance. In order to avoid this difficulty and also to improve efficiency in the event of rapid convergence, a test based on only two trial vectors is applied first.

The rapid convergence test is based on the fact that the sweeping procedures, Section 10.4.2.4, guarantee the orthogonality of very close eigenvalues. It is only necessary to avoid contamination with eigenvectors that are not in the cluster. Thus if

$$\left|\frac{\lambda_2 - \lambda_1}{\lambda_1}\right| = \left|\frac{\Lambda_2 - \Lambda_1}{\Lambda_1 + \lambda_o}\right| < \gamma \quad , \tag{36}$$

where γ is a small parameter built into the program, then it is not required that Equation 35 be satisfied.

If the criterion obtained by multiplying Equations 35 and 36 together, namely

$$\left|\frac{\Lambda_2 - \Lambda_1}{\Lambda_1 + \lambda_o}\right| \sqrt{\delta_n} < A\varepsilon\gamma \quad , \tag{37}$$

is satisfied, then either Equation 35 or 36, or both, are satisfied. From Equation 34

$$\left|\frac{\Lambda_2 - \Lambda_1}{\Lambda_1 + \lambda_o}\right| \sqrt{\delta_n} = \frac{\left[(\delta u_n)^T M(\delta u_n)\right]^{1/2}}{\left|1 + \frac{\lambda_o}{\Lambda_1}\right|} \quad . \tag{38}$$

The rapid convergence test is, therefore

$$\eta = \left[(\delta u_n)^T M \delta u_n\right]^{1/2} < A\varepsilon\gamma \left|1 + \frac{\lambda_c}{\Lambda_1}\right| \quad . \tag{39}$$

Λ_1 is evaluated from Equation 26. The test requires only two successive trial vectors, and it is performed as often as possible. The value of γ stored in the program is .01.

10.4.2.3 Change of Shift Point

The rate of convergence of the iteration algorithm can be estimated from the parameters, Λ_1, Λ_2, and δ_n. From Equations 19 and 20, the estimated proportion of the second eigenvector in the (n+k)th trial eigenvector is

$$\sqrt{\delta_{n+k}} = \left(\frac{\Lambda_1}{\Lambda_2}\right)^k \sqrt{\delta_n} \quad . \tag{40}$$

If the (n+k)th trial eigenvector satisfies the convergence criterion, Equation 35, then

$$\left(\frac{\Lambda_1}{\Lambda_2}\right)^k < \frac{A\varepsilon}{\sqrt{\delta_n}} \quad , \tag{41}$$

and the estimate of the required number of additional iterations, beyond the nth, is

$$k > \frac{\log\left(\frac{\sqrt{\delta_n}}{A\varepsilon}\right)}{\log\left|\frac{\Lambda_2}{\Lambda_1}\right|} > k - 1 \quad . \tag{42}$$

ORIGINAL PAGE IS
OF POOR QUALITY

If $|\Lambda_2/\Lambda_1|$ is nearly equal to unity, the required number of iterations may be very large.

Suppose, for example, that

$$\varepsilon = 10^{-6}, \quad A = 10^{-1}, \quad \delta_n = 10^{-4}, \quad \frac{\Lambda_2}{\Lambda_1} = 1.10 \ . \quad \text{Then}$$

$$k > \frac{\log(10^5)}{\log(1.1)} = 119 \quad .$$

If, however, the shift point is changed to the current estimate of the eigenvalue, $\Lambda_{1,n} + \lambda_o$, the number of additional iterations will usually be quite small. The error in the calculation of Λ_1 from Equation 26 is of the order of δ_n, as may be seen by comparing Equation 26 and Equation 27. Thus assuming the new shift point to be located at $\Lambda_1(1-\delta_n)$, the ratio of the first and second eigenvalues from the new shift point is

$$\frac{\Lambda_2'}{\Lambda_1'} = \frac{\Lambda_2 - \Lambda_1(1-\delta_n)}{\Lambda_1 - \Lambda_1(1-\delta_n)} = \frac{\Lambda_2 - \Lambda_1}{\delta_n \Lambda_1} \quad . \tag{43}$$

Using this result in the previous example, the estimated number of iterations from the new shift point is

$$k' = \frac{\log(10^5)}{\log(10^3)} = 1.6 \quad .$$

In many cases the time required to complete the estimated number of iterations from the old shift point may exceed the time required to make a triangular decomposition and to complete the iteration from the new shift point. In such cases a decision to shift is made at the earliest time that it can reliably be made.

Let T_d be the time for one triangular decomposition and let T_i be the time for one iteration. Then, if

$$T_d < (k - k')T_i \quad , \tag{44}$$

a new shift point is selected. The minimum estimate of k' is 2, to permit testing for convergence after the shift. The relative times for decomposition and iteration depend on the bandwidth of the dynamic matrix. As a first approximation

$$\frac{T_d}{T_i} \simeq \frac{B}{3} \quad , \tag{45}$$

668

where B is the semibandwidth. Thus, as a first approximation, a new shift point should be selected if

$$B < 3(k - k') \quad . \tag{46}$$

In practice accurate values of T_d and T_i are available to the computer because at least one triangular decomposition and several iterations have already been performed.

Although the test described above will predict the rate of convergence, it is not foolproof for two reasons. The first is the requirement that the proportion in the trial vectors of all eigenvectors except the two lowest be very small. The second is that there must be a sufficient proportion of the second eigenvector to permit reasonably reliable calculation of Λ_2. A check on the convergence criteria is provided by examining the series of values of Λ_1 and Λ_2. If the differences between successive values decrease monotonically, we are assured of uniform convergence. On the other hand, a sudden jump in the value of Λ_2 will indicate that the limit of digital precision has been reached.

Two additional tests are performed in NASTRAN to improve reliability.

Define the normalized increments in the approximate eigenvalues from successive iterations

$$\bar{h}_{1,n} = \frac{\lambda_{1,n} - \lambda_{1,n-1}}{R_o} = \frac{1}{R_o}(\Lambda_{1,n} - \Lambda_{1,n-1}) \quad , \tag{47}$$

and

$$\bar{h}_{2,n} = \frac{\lambda_{2,n} - \lambda_{2,n-1}}{\lambda_o} = \frac{(\Lambda_{2,n} - \Lambda_{2,n-1})}{\lambda_o} \quad , \tag{48}$$

where R_o is a normalizing factor to be specified later (see Equation 69). The shift test, Equation 44, is applied provided that

$$|\bar{h}_{1,n}| < \varepsilon_1 \quad . \tag{49}$$

The criterion ε_1 is not made very severe. The reason for the criterion is that, if we decide to shift, we wish some assurance that the new shift point is substantially better than the original shift point.

The second test relates to the reliability of Λ_2. If

$$\varepsilon_2 > |\bar{h}_{2,n}| > |\bar{h}_{2,n-1}| \quad , \tag{50}$$

ORIGINAL PAGE IS OF POOR QUALITY

then $\Lambda_{2,n-1}$ will be used in all subsequent calculations of δ from Equation 34. Otherwise continue to compute Λ_2 at each step. Note that the ratio Λ_2/Λ_1 from Equation 33 is no longer required after the ε_2 test has failed.

10.4.2.4 Sweeping of Previously Found Eigenvectors

The trial vectors must be swept to eliminate contributions due to previously found eigenvalues that are closer to the shift point than the current eigenvalue. The algorithm is developed as follows. Let the unswept trial vector be $\overline{u}_n$ which, from Equation 9, can be represented as:

$$\overline{u}_n = \sum_{i=1}^{m} \alpha_{i,n}\phi_i + u_n \quad , \qquad (51)$$

where ϕ_i, i = 1,2...m, are exact values of the previously extracted eigenvalues and u_n is the swept trial vector which will include only contributions from the remaining modes. The computed values for previously extracted eigenvectors are

$$\overline{\phi}_i = \frac{u_{i,N}}{\sqrt{u_{i,N}^T M u_{i,N}}} \quad , \qquad (52)$$

where $u_{i,N}$ is the last (N^{th}) trial vector obtained in the iteration for λ_i.

If $\overline{\phi}_i$ were an exact eigenvector, then, from the mass-orthogonality properties,

$$\alpha_{i,n} = \overline{\phi}_i^T M \overline{u}_n \quad , \qquad (53)$$

and

$$u_n = \overline{u}_n - \sum_i \left(\overline{\phi}_i^T M \overline{u}_n \right) \overline{\phi}_i \quad , \qquad (54)$$

which is the sweeping algorithm.

Let us examine the error in the process. Suppose

$$\overline{\phi}_i = \phi_i + \varepsilon\phi_j \quad , \qquad (55)$$

where ϕ_i and ϕ_j are exact eigenvectors and $\overline{\phi}_i$ is the eigenvector that is currently being extracted. Also let

$$\overline{u}_n = \alpha_{i,n}\phi_i + \alpha_{j,n}\phi_j \quad . \qquad (56)$$

Thus,

$$(\bar{\phi}_i^T M \bar{u}_n) = \alpha_{i,n} + \varepsilon\alpha_{j,n} \quad , \tag{57}$$

and, considering only one previously swept eigenvector in Equation 54

$$\begin{aligned} u_n &= \alpha_{i,n}\phi_i + \alpha_{j,n}\phi_j - (\alpha_{i,n} + \varepsilon\alpha_{j,n})(\phi_i + \varepsilon\phi_j) \\ &= \left(\alpha_{j,n}(1 - \varepsilon^2) - \varepsilon\alpha_{i,n}\right)\phi_j - \varepsilon\alpha_{j,n}\phi_i \quad . \end{aligned} \tag{58}$$

Redefine the coefficients of ϕ_i and ϕ_j in Equation 58

$$u_n = \bar{\alpha}_{j,n}\phi_j + \bar{\alpha}_{i,n}\phi_i \quad . \tag{59}$$

The ratio of the relative amplitudes of eigenvector components in the unswept vectors obtained after the next iteration is, for $\varepsilon << 1$,

$$\frac{\alpha_{i,n+1}}{\alpha_{j,n+1}} = \frac{\Lambda_j}{\Lambda_i}\frac{\bar{\alpha}_{i,n}}{\bar{\alpha}_{j,n}} \cong -\varepsilon\frac{\Lambda_j}{\Lambda_i} \quad . \tag{60}$$

Convergence to ϕ_j rather than to ϕ_i requires that

$$\left|\varepsilon\frac{\Lambda_j}{\Lambda_i}\right| < 1 \quad . \tag{61}$$

If all previous eigenvectors are reasonably accurate, we need only to be concerned for cases in which $\Lambda_j >> \Lambda_i$. This is, however, precisely the condition that exists at the shifted shift point, λ_o'. Thus, λ_o' should not be used for obtaining additional eigenvalues.

Furthermore, if an eigenvalue is found to be very close to the original shift point, then the shift point should be changed in the search for other eigenvalues. The criterion for changing starting points (original shift points) may be written

$$\varepsilon\left|\frac{R_o}{\lambda_o' - \lambda_1}\right| > \varepsilon_3 \quad , \tag{62}$$

where ε is the coefficient specified in the mass-orthogonality test, Equation 13, λ_1 is the first eigenvalue found from the starting point λ_o, R_o is the distance from λ_o to the farthest eigenvalue to be extracted from λ_o, and ε_3 is a safety factor. Suppose, for example, $\varepsilon = 10^{-6}$ and $\varepsilon_3 = 10^{-2}$. Then the starting point should be changed if

$$\left|\frac{\lambda_o - \lambda_1}{R_o}\right| < 10^{-4} \quad . \tag{63}$$

Another conclusion to be derived from Equation 62 is that the trial vectors should be swept at each iteration. If, for example, trial vectors were swept every k iterations, proper convergence would require that

$$\varepsilon\left|\frac{R_o}{\lambda_o - \lambda_1}\right|^k < 1 \quad , \tag{64}$$

which, if k were larger than about 3 or 4, would require continuous changing of starting points.

10.4.2.5 Initial Trial Vectors and the Extraction of Multiple Eigenvectors

The choice of a trial vector for finding the first eigenvalue from any starting point is clearly arbitrary. The vector used in NASTRAN has a broad spectrum with regard to spatial distribution in order to provide efficient extraction of eigenvalues in the complete wavelength range.

There is an advantage in finding successive eigenvalues from the same starting point, in using one of the trial vectors employed in a late stage of the previous iteration. The reason for the advantage is that such a vector, after having been swept, is quite "rich" in the eigenvector that is next closest to the starting point and will, therefore, converge more rapidly. There is a difficulty, however, in that, if the previous eigenvalue is a multiple eigenvalue, a return to it will probably be prevented since the swept trial vector contains no component of the multiple vector(s). If the process of transforming trial vectors is repeated often enough, however, the process will eventually return to the multiple eigenvalue by virtue of the growth of round-off error. Unfortunately this cannot be relied upon. The safer course, which is to use a new initial trial vector for each eigenvalue, will require more iterations since there is no "enrichment".

The following scheme, in which we might have our cake and eat it too, is used in NASTRAN:

1. Find the first eigenvalue from a starting point with an arbitrarily selected trial vector.

2. Use the next to last trial vector obtained at the starting point during the first iteration sequence as the initial trial vector for the second iteration sequence. This will ensure a very large proportion of the next closest eigenvector. Since we are

required to save two vectors at every iteration step in order to check convergence, this should be no burden on storage requirements. Furthermore, since all calculations are done in double precision, and only single-precision convergence of eigenvectors can be required, the swept vector should not contain much random noise.

3. Continue until an eigenvalue is found that is outside the prescribed range of the starting point (see Section 10.4.2.6).
4. At this stage, use a new arbitrarily selected initial vector and find one eigenvalue.
5. If the eigenvalue found in step 4 is outside the range of the starting point, go to a new starting point. If it is not outside the range, repeat steps 2, 3 and 4 until the eigenvalue found in step 4 is outside the range. The maximum number of times that steps 2, 3 and 4 will be performed is equal to the multiplicity of the eigenvalue with greatest multiplicity.

An exception to the above sequence occurs if step 1 yields an eigenvalue outside the range. In that case, the program goes directly to a new starting point.

10.4.2.6 Distribution of Starting Points

For Vibration Analysis the user specifies:

1. The maximum frequency of interest, f_{max}.
2. The minimum frequency of interest, f_{min}.
3. The estimated number of modes in the range, N_e.
4. The desired number of modes, N_d, to be computed.

For Buckling Analysis the user specifies:

1. The maximum eigenvalue of interest, λ_{max}.
2. The minimum eigenvalue of interest, λ_{min}, which may be negative.
3. The estimated number of modes in the range, N_e.
4. The desired number of modes, N_d, to be computed.

For Vibration Analysis, the eigenvalue

$$\lambda = (2\pi f)^2 = \omega^2 \quad . \tag{65}$$

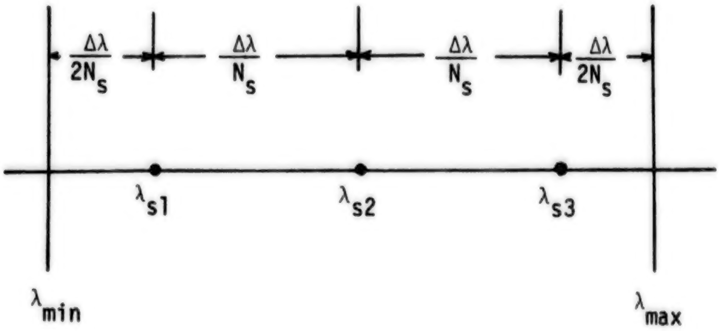
If the desired number of modes, N_d, is not specified, then all modes in the range will be found provided that there are not more than $3N_e$ modes. If the desired number of eigenvalues is less than the expected number, those that are closest to zero will be computed.

Starting points are distributed on the assumption that the eigenvalues are uniformly distributed with respect to λ, (frequency-squared), and that about six eigenvalues will be found from each starting point.

The number of starting points, N_s, is selected such that

$$N_s - 1 < \frac{N_e}{6} < N_s \quad , \tag{66}$$

and they are distributed as follows.


$\frac{\Delta\lambda}{2N_s}$
$\frac{\Delta\lambda}{N_s}$
$\frac{\Delta\lambda}{N_s}$
$\frac{\Delta\lambda}{2N_s}$
λ_{s1}
λ_{s2}
λ_{s3}
λ_{min}
λ_{max}

where $\Delta\lambda = \lambda_{max} - \lambda_{min} = (2\pi f_{max})^2 - (2\pi f_{min})^2$. Thus the nth starting point,

$$\lambda_{sn} = \lambda_{min} + \frac{n - \frac{1}{2}}{N_s} \Delta\lambda \quad . \tag{67}$$

The search-ranges for the starting points are selected to overlap each other by 10%. Thus, the search-range for shifted eigenvalues, $\Lambda = \lambda - \lambda_{sn}$, is

$$-R_o < \Lambda < +R_o \quad , \tag{68}$$

where

$$R_o = .55 \frac{\Delta\lambda}{N_s} \quad . \tag{69}$$

R_o is also used both in deciding whether or not to relocate starting points, Equation 62, and in the ε_1 preliminary convergence test, Equation 47.

10.4.2.7 Termination

The process of finding eigenvalues can be terminated for any of the following reasons:

1. All of the eigenvalues that the problem contains have been found.
2. All of the eigenvalues in the desired range, $\lambda_{min} < \lambda < \lambda_{max}$, have been found.
3. The desired number of eigenvalues, N_d, have been found.
4. The number of eigenvalues that have been found is larger by some factor than the number estimated to be in the desired range.

Reason 1 is imperative because, if the iteration process is continued, it will converge on eigenvalues that it has already found and give false results. A reasonable, although not foolproof, test for whether or not all (finite) eigenvalues have been found is to compute the total number of eigenvalues from

$$N_t = N_{eq} - N_o - N_{oM} \quad , \tag{70}$$

where

N_{eq} = number of rows in the dynamic matrix.

N_o = number of zero eigenvalues prescribed by the user, and found by separate procedures.

N_{oM} = number of diagonal terms of the mass matrix that are equal to zero.

The test is not foolproof because the mass matrix can be singular even if none of the diagonal terms are zero. The detection of general singularity is regarded as too difficult to be worthwhile for the present application.

Reason 2 is detected quite simply by applying the range test to the last starting point.

The recommended value for the factor mentioned in reason 4 is 3.0.

Once the process of extracting eigenvalues has terminated, all of the eigenvalues and eigenvectors that have been found (and not just the desired number or those within the desired range) are prepared for output. If eigenvalues are to be used in a modal solution they will all be positive and only the desired number closest to the origin (if that number has been specified), or the total number within the specified frequency range, whichever is less, are used.

10.4.3 Summary of Procedures for Real Eigenvalue Analysis

Flow diagrams illustrating recommended procedures are shown in Figures 1, 2 and 3. The procedures involved in each block are summarized below.

1. Compute Distribution of Starting Points (see Section 10.4.2.6)

a. Select number of starting points, N_s, by:

$$N_s - 1 < \frac{N_e}{6} < N_s \quad , \tag{1}$$

where N_e is the estimated number of eigenvalues in the frequency range of interest.

b. Locate starting points by:

$$\lambda_{sn} = \lambda_{min} + \frac{n - \frac{1}{2}}{N_s} \Delta\lambda \quad , \tag{2}$$

where

$$\Delta\lambda = \lambda_{max} - \lambda_{min} = (2\pi f_{max})^2 - (2\pi f_{min})^2 \quad . \tag{3}$$

2. Select a Starting Point

The first starting point is λ_{s1}, the second starting point is λ_{s2}, etc., until all starting points are used up or until the desired number, N_d, of roots has been found.

3. Select an Arbitrary Starting Vector

Each arbitrary starting vector is distinct from all preceding starting vectors. The vector should have a broad spectrum with regard to spatial distribution (i.e., not concentrated in the long wavelength or in the short wavelength range).

4. Compute One Eigenvalue and One Eigenvector

See flow diagram for this block, Figure 2.

4.1 Decompose Dynamic Matrix

The matrix to be decomposed is $[K - \lambda_o M]$ which is real and symmetric and where λ_o is either a starting point, λ_s, a moved starting point, λ_s', or a shifted shift point, λ_o'. Real arithmetic without pivoting is used. Partial pivoting is available on an optional basis.

4.2 One Vector Iteration (see Sections 10.4.2.1 and 10.4.2.4)

The iteration algorithm is:

$$[K - \lambda_o M]\{w_n\} = [M]\{u_{n-1}\} \equiv \{F_{n-1}\} \quad , \tag{4}$$

$$\{\bar{u}_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{5}$$

$$\{u_n\} = \{\bar{u}_n\} - \sum_i [\phi_i{}^T M \bar{u}_n]\{\phi_i\} \; , \tag{6}$$

where C_n is equal to the element of $\{w_n\}$ with largest magnitude and $\{\phi_i\}$ is a normalized eigenvector, previously found:

$$\{\phi_i\} = \frac{\{u_{i,N}\}}{\sqrt{u_{i,N}{}^T M u_{i,N}}} \quad . \tag{7}$$

$\{u_{i,N}\}$ is the last vector found in iterating for the ith eigenvalue. The sum on (i) in Equation 6 extends over all eigenvectors previously found, including rigid body modes found in separate procedures. The sweeping operation, Equation 6, is also applied to starting vectors, $\{\bar{u}_o\}$.

4.3 Convergence Tests (see Section 10.4.2.2)

See flow diagram for this block, Figure 3.

a. The quantities Λ_1, Λ_2, η, and δ are computed as follows.

1. Form
$$\{F_n\} = [M]\{u_n\} \quad , \tag{8}$$
which is also required in the next iteration (see Equation 4).

2. Compute
$$\alpha_n = (\{u_n\}^T\{F_n\})^{1/2} \quad . \tag{9}$$

3. Compute
$$\{\delta u_n\} = \frac{\{u_n\}}{\alpha_n} - \frac{\{u_{n-1}\}}{\alpha_{n-1}} \quad . \tag{10}$$

4. Compute
$$\{\delta F_n\} = \frac{\{F_n\}}{\alpha_n} - \frac{\{F_{n-1}\}}{\alpha_{n-1}} \quad . \tag{11}$$

Then

5. $$\Lambda_1 = \frac{1}{C_n}\frac{\alpha_{n-1}}{\alpha_n} \quad . \tag{12}$$

6.

$$\eta = (\{\delta u_n\}^T\{\delta F_n\})^{1/2} \quad , \qquad , \tag{13}$$

used in the rapid convergence test.

7.

$$\frac{\Lambda_2}{\Lambda_1} = \frac{\{\delta u_n\}^T\{\delta F_{n-1}\}}{\{\delta u_n\}^T\{\delta F_n\}} \quad . \tag{14}$$

8.

$$\delta = \frac{\{\delta u_n\}^T\{\delta F_n\}}{\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2} \quad . \tag{15}$$

except that the maximum value of $\left(1 - \frac{\Lambda_2}{\Lambda_1}\right)^2$ is limited to 10.0.

b. In the rapid convergence test,

$$\eta < A\epsilon\gamma \left| 1 + \frac{\lambda_o}{\Lambda_1} \right| \quad , \tag{16}$$

ϵ = criterion specified by the user for the mass-orthogonality check,

γ = close root criterion (= 10^{-2}),

A = safety factor (= 10^{-1}).

The default value of ϵ is 10^{-4}.

c. The test sequence

$$\eta_{n-1} < 10^{-6} \quad , \tag{17}$$

$$\eta_n > 1.01\, \eta_{n-1} \quad , \tag{18}$$

is introduced to provide for situations in which round-off error dominates the convergence criteria. These tests were rot discussed in Section 10.4.2.

d. The ϵ_2 test is:

If $\epsilon_2 > |\bar{h}_{2,n}| > |\bar{h}_{2,n-1}|$: Fail . (19)

Otherwise: Pass.

where, using successive values of Λ_2 computed by Equation 14,

$$\bar{h}_{2,n} = \frac{\Lambda_{2,n} - \Lambda_{2,n-1}}{\lambda_0} \quad . \qquad (20)$$

The value of ε_2 stored in the program is .02.

If the ε_2 test fails, the value of Λ_2 is set equal to $\Lambda_{2,n-1}$ in calculating δ (Equation 15) and is not changed until the shift point is changed or an eigenvalue is found.

4.4 ε_1 Test (see Section 10.4.2.3)

The ε_1 test is

$$\text{if} \qquad \left| \frac{\Lambda_{1,n} - \Lambda_{1,n-1}}{R_0} \right| < \varepsilon_1 \ : \ \text{Pass . Proceed to 4.5.} \qquad (21)$$

Otherwise: Fail. Return to 4.2.

Where

$$R_0 = .55 \frac{\Delta\lambda}{N_s} \quad . \qquad (22)$$

The value of ε_1 stored in the program is 10^{-3}. The ε_1 test is also applied as part of the rapid convergence procedure, see Figure 2.

4.5 Shift Decision (see Section 10.4.2.3)

The criterion for shifting is

$$\text{if} \qquad T_d < (k - 2)T_i \ , \qquad (23)$$

or if for two successive iterations

$$\left| \frac{\Lambda_2}{\Lambda_1} \right| \leq 1 \ : \ \text{Shift.}$$

Otherwise: Continue to iterate.

Where

T_d = estimated time for one triangular decomposition,

T_i = estimated time for one vector iteration,

k = estimated number of additional iterations.

k is evaluated from

$$k = \frac{\ln\sqrt{\frac{|\delta|}{A\varepsilon}}}{\ln\left|\frac{\Lambda_2}{\Lambda_1}\right|} \quad . \tag{24}$$

4.6 Change Shift Point

Change shift point to

$$\lambda_o' = \lambda_o + \Lambda_1 \quad , \tag{25}$$

where Λ_1 is computed by Equation 12.

4.7 Reset Iteration Counter

Set $n = 0$, and also remove the flag that the ε_2 test has failed.

5. Are We Done?

The answer is yes if:

a. The total desired number of eigenvalues, N_d, have been found, or

b. The number found exceeds three times the estimated number, N_e, or

c. The number of eigenvalues found equals the total number in the problem. The total number in the problem is computed from

$$N_t = N_{eq} -- N_o - N_{oM} \quad , \tag{26}$$

where

N_{eq} = number of rows in the dynamic matrix,

N_o = number of zero eigenvalues prescribed by the user and found by separate procedures,

N_{oM} = number of diagonal terms of the mass matrix that are equal to zero.

6. Is Eigenvalue Outside Range? (see Section 10.4.2.6)

The test is:

$$-R_o < \lambda_i - \lambda_s < +R_o \quad : \text{ Inside range.} \tag{27}$$

Otherwise: Outside range.

7. Select Next-to-Last Trial Vector as Starting Vector (see Section 10.4.2.5)

a. If there was no shift in finding the previous eigenvalue (i.e., if Equation 23 has never been satisfied) set:

$$\bar{u}_{i+1,o} = u_{i,N-1} \quad . \tag{28}$$

b. If there were one or more shifts, set $\bar{u}_{i+1,o}$ equal to last trial vector before the first shift.

8. Too Close to Starting Point? (see Section 10.4.2.4)

The criterion is:

$$\varepsilon \left| \frac{R_o}{\lambda_i - \lambda_s} \right| > \varepsilon_3 \quad : \text{ Too close} \quad , \tag{29}$$

where

ε = criterion used in mass orthogonality test,

λ_i = eigenvalue just found,

λ_s = starting point,

R_o = range, see Equation 22.

The value for ε_3 stored in the program is .05.

9. Move Starting Point

Change starting point location to

$$\lambda_s' = \lambda_s \pm .02\, R_o \quad , \tag{30}$$

where the (+) or (-) is selected to move the starting point away from the eigenvalue.

10. Is This the First Eigenvalue Found Since the Arbitrary Starting Point Vector Was Changed?

See Section 10.4.2.5 on Initial Trial Vectors.

11. Last Starting Point?

If the answer is yes, we are done. If the answer is no, select a new starting point.

10.4.4 Theory for Complex Eigenvalue Analysis

10.4.4.1 Iteration Algorithm

The problem is to find the eigenvalues and eigenvectors for

$$[Mp^2 + Bp + K]\{u\} = 0 \quad , \tag{1}$$

where $[M]$, $[B]$, and $[K]$ may be real or complex, symmetric or nonsymmetric, singular or nonsingular. Eigenvalues may be multiple. All the eigenvalues within specified regions of the p-plane are to be found.

Define the velocity vector

$$\{v\} = p\{u\} \quad . \tag{2}$$

Equation 1 may then be rewritten in partitioned form as

$$p\left[\begin{array}{c|c} M & 0 \\ \hline 0 & I \end{array}\right]\left\{\begin{array}{c} v \\ \hline u \end{array}\right\} + \left[\begin{array}{c|c} B & K \\ \hline -I & 0 \end{array}\right]\left\{\begin{array}{c} v \\ \hline u \end{array}\right\} = 0 \quad . \tag{3}$$

Let

$$p = \lambda_o + \Lambda \quad , \tag{4}$$

where the constant λ_o is called the shift point. Substituting Equation 4 into Equation 3:

$$\left[\begin{array}{c|c} -B - M\lambda_o & -K \\ \hline I & -\lambda_o I \end{array}\right]\left\{\begin{array}{c} v \\ \hline u \end{array}\right\} = \Lambda\left[\begin{array}{c|c} M & 0 \\ \hline 0 & I \end{array}\right]\left\{\begin{array}{c} v \\ \hline u \end{array}\right\} \quad , \tag{5}$$

or, assuming for the moment that $[M]$ is nonsingular:

$$\left[\begin{array}{c|c} -M^{-1}B - \lambda_o I & -M^{-1}K \\ \hline I & -\lambda_o I \end{array}\right]\left\{\begin{array}{c} v \\ \hline u \end{array}\right\} = \Lambda\left\{\begin{array}{c} v \\ \hline u \end{array}\right\} \quad . \tag{6}$$

PAGE 10.4-23B INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

Equation 6 is an eigenvalue problem in standard form, i.e.,

$$[A]\{x\} = \Lambda\{x\} \quad , \tag{7}$$

where

$$\{x\} = \left\{\begin{array}{c} v \\ \hline u \end{array}\right\} \quad .$$

The standard iteration algorithm for the inverse power method is

$$[A]\{x_n\} = \frac{1}{C_n}\{x_{n-1}\} \quad , \tag{8}$$

where C_n is a normalizing factor to be determined. Convergence of the algorithm is discussed in Section 10.4.4.3.

An alternate form of the iteration algorithm corresponding to Equation 5 is:

$$\left[\begin{array}{c|c} -B - \lambda_o M & -K \\ \hline I & -\lambda_o I \end{array}\right] \left\{\begin{array}{c} v_n \\ \hline u_n \end{array}\right\} = \frac{1}{C_n} \left[\begin{array}{c|c} M & 0 \\ \hline 0 & I \end{array}\right] \left\{\begin{array}{c} v_{n-1} \\ \hline u_{n-1} \end{array}\right\} \quad . \tag{9}$$

A form that is more convenient for computation is obtained as follows: From the second row of Equation 9, suppressing matrix brackets for conciseness,

$$v_n = \lambda_o u_n + \frac{1}{C_n} u_{n-1} \quad . \tag{10}$$

Substitute Equation 10 into the top row of Equation 9:

$$-(B + \lambda_o M)(\lambda_o u_n + \frac{1}{C_n} u_{n-1}) - K u_n = \frac{1}{C_n} M v_{n-1} \quad , \tag{11}$$

or

$$(\lambda_o{}^2 M + \lambda_o B + K) u_n = -\frac{1}{C_n}\Big((B + \lambda_o M) u_{n-1} + M v_{n-1}\Big) \quad . \tag{12}$$

Let

$$w_n = C_n u_n \quad . \tag{13}$$

The complete algorithm is, then

$$
\begin{aligned}
(\lambda_o^2 M + \lambda_o B + K) w_n &= -(B + \lambda_o M) u_{n-1} - M v_{n-1} \quad , \\
u_n &= \frac{1}{C_n} w_n \quad , \qquad (14) \\
v_n &= \lambda_o u_n + \frac{1}{C_n} u_{n-1} \quad .
\end{aligned}
$$

C_n is selected equal to the element of w_n with largest magnitude. Note that triangular decomposition of the dynamic matrix $[p^2 M + pB + K]$ is required at the point $p = \lambda_o$.

10.4.4.2 Orthogonality Properties for Nonsymmetric Matrices

Development of orthogonality properties is required in order to prove convergence and to derive sweeping techniques. Consider the pair of eigenvalue problems

$$Ax = \Lambda x \quad , \qquad (15)$$

$$A^T y = \Lambda y \quad , \qquad (16)$$

where A is a general, nonsymmetric, real or complex matrix. The eigenvalues of Equation 15 are the zeroes of the determinant $|A - \Lambda I|$. The eigenvalues of Equations 15 and 16 are the same because

$$|A - \Lambda I| = |(A^T - \Lambda I)^T| = |(A^T - \Lambda I)| \quad . \qquad (17)$$

The eigenvectors of Equations 15 and 16 are not, however, identical. The eigenvectors satisfy

$$(A - \Lambda_i)\phi_i = 0 \quad , \qquad (18)$$

$$(A^T - \Lambda_j)\bar{\bar{\phi}}_j = 0 \quad . \qquad (19)$$

ϕ_i is called a right eigenvector of A and $\bar{\bar{\phi}}_j$ is called a left eigenvector of A because, transposing Equation 19,

$$\bar{\bar{\phi}}_j^T(A - \Lambda_j) = 0 \tag{20}$$

The left and right eigenvectors are orthogonal for $\Lambda_i \neq \Lambda_j$. To show this, premultiply Equation 18 by $\bar{\bar{\phi}}_j^T$

$$\bar{\bar{\phi}}_j^T A\phi_i - \Lambda_i \bar{\bar{\phi}}_j^T \phi_i = 0 \quad , \tag{21}$$

or, using Equation 20

$$(\Lambda_j - \Lambda_i)\bar{\bar{\phi}}_j^T \phi_i = 0 \quad , \tag{22}$$

so that, for $\Lambda_i \neq \Lambda_j$

$$\bar{\bar{\phi}}_j^T \phi_i = 0 \quad , \tag{23}$$

which is known as the property of biorthogonality.

Equation 23 will be useful in proving convergence of the iteration algorithm. It is not useful for sweeping procedures because transposition of the A matrix in Equation 6 leads to a form that is extremely awkward for computation. A more convenient form of orthogonality property is developed as follows. Let ϕ_i be an eigenvector satisfying

$$(p_i^2 M + p_i B + K)\phi_i = 0 \quad , \tag{24}$$

and $\bar{\bar{\phi}}_j$ be an eigenvector satisfying

$$\bar{\bar{\phi}}_j^T(p_j^2 M + p_j B + K) = 0 \quad . \tag{25}$$

Premultiply Equation 24 by $p_j\bar{\bar{\phi}}_j^T$, postmultiply Equation 25 by $p_i\phi_i$, and subtract;

$$p_j\bar{\bar{\phi}}_j^T(p_i^2 M + p_i B + K)\phi_i - \bar{\bar{\phi}}_j^T(p_j^2 M + p_j B + K)p_i\phi_i = 0 \quad . \tag{26}$$

Combine terms:

$$(p_i - p_j)\bar{\bar{\phi}}_j^T[p_i p_j M - K]\phi_i = 0 \quad , \tag{27}$$

so that for $p_i \neq p_j$

$$\bar{\bar{\phi}}_j^T[p_i p_j M - K]\phi_i = 0 \quad . \tag{28}$$

Postmultiply Equation 25 by ϕ_i, and find that

$$-\bar{\bar{\phi}}_j^T K \phi_i = \bar{\bar{\phi}}_j^T(p_j^2 M + p_j B)\phi_i \quad . \tag{29}$$

Substitute this result into Equation 28:

$$(p_i + p_j)\bar{\bar{\phi}}_j^T M \phi_i + \bar{\bar{\phi}}_j^T B \phi_i = 0 \quad , \tag{30}$$

which is the desired result. Equation 30 is used in the sweeping algorithm. Note that Equation 30 is the analog of the mass orthogonality relationship (Equation 5 of Section 10.4.2.1) extended to complex unsymmetric matrices, including the presence of damping.

10.4.4.3 Proof of Convergence for the Iteration Algorithm

Consider the vector

$$\{x_n\} = \left\{\begin{array}{c} v_n \\ \hdashline u_n \end{array}\right\} \quad , \tag{31}$$

introduced in Equation 7. Expand $\{x_n\}$ and $\{x_{n-1}\}$ in terms of eigenvectors

$$x_n = \sum_i \alpha_{i,n}\phi_i \quad , \tag{32}$$

$$x_{n-1} = \sum_i \alpha_{i,n-1}\phi_i \quad . \tag{33}$$

Then, substituting into Equation 8 and premultiplying by $\bar{\bar{\phi}}_j^T$ which is the left eigenvector satisfying Equation 20,

$$\bar{\bar{\phi}}_j^{\,T} A \sum_i \alpha_{i,n}\phi_i = \frac{1}{C_n}\sum_i \alpha_{i,n-1}\bar{\bar{\phi}}_j^{\,T}\phi_i \quad , \qquad (34)$$

or, using Equation 18,

$$\sum_i \alpha_{i,n}\Lambda_i\bar{\bar{\phi}}_j^{\,T}\phi_i = \frac{1}{C_n}\sum_i \alpha_{i,n-1}\bar{\bar{\phi}}_j^{\,T}\phi_i \quad , \qquad (35)$$

and using the orthogonality property, Equation 23,

$$\alpha_{j,n} = \frac{1}{\Lambda_j C_n}\alpha_{j,n-1} \quad . \qquad (36)$$

Thus the eigenvector with lowest value of Λ_j (i.e., with eigenvalue closest to the shift point) increases more rapidly than the others. The above proof of convergence is curious in that it requires the postulation of a second eigenvalue problem. It would be more satisfying if such were not required.

A more serious objection to the proof of convergence is the assumption that the mass matrix, [M], is nonsingular, which assumption is required in the reduction of the eigenvalue problem to standard form (see Section 10.4.4.1). The objection may be resolved by examining what happens as [M] approaches singularity. What happens, in fact, is that one (or more if the defect of [M] is greater than one) of the eigenvalues approaches infinity. Convergence to the corresponding eigenvector from any (finite) shift point is impossible. Otherwise the problem is not exceptional, and it may be concluded that convergence to finite eigenvalues will exist even in the limit when [M] is singular.

It will be noted that the rate of convergence of eigenvectors is the same as the rate of convergence in the real eigenvalue problem (see Equation 12 of Section 10.4.2.1). Thus arguments concerning the rate of convergence of eigenvectors in the real eigenvalue problem also apply to the complex eigenvalue problem. In particular, the test used in the decision to change the shift point is the same.

10.4.4.4 Sweeping of Previously Found Eigenvectors

Before each iteration, components of previously found eigenvectors are removed from the vector $\{x_n\} = \left\{\begin{array}{c} v_n \\ \hline u_n \end{array}\right\}$ in order to permit convergence to a new eigenvector. The procedure is as follows. Represent $\{x_n\}$ as

$$\left\{\begin{array}{c} v_n \\ \hline u_n \end{array}\right\} = \sum_{j=1}^{m} \alpha_{j,n} \left\{\begin{array}{c} p_j\phi_j \\ \hline \phi_j \end{array}\right\} + \left\{\begin{array}{c} v_n \\ \hline u_n \end{array}\right\}_p \quad , \tag{37}$$

where ϕ_j is a previously accepted approximation to an eigenvector, where the sum extends over all m previously extracted eigenvectors, and where the last term on the right is the "purified" trial vector. The problem is to evaluate $\alpha_{j,n}$. For this purpose it is helpful to form the scalar matrix product

$$S_{jn} = \bar{\bar{\phi}}_j^T[p_jMu_n + Mv_n + Bu_n] \quad , \tag{38}$$

where $\bar{\bar{\phi}}_j$ is a previously accepted left eigenvector. Expanding u_n and v_n in terms of eigenvectors as in Equation 37, except that m ranges over all eigenvalues,

$$S_{jn} = \bar{\bar{\phi}}_j^T \sum_{i=1}^{m} \alpha_{i,n}\Big((p_i + p_j)M\phi_i + B\phi_i\Big) \quad , \tag{39}$$

or using the orthogonality relationship, Equation 30,

$$S_{jn} = \alpha_{j,n}\bar{\bar{\phi}}_j^T(2p_jM + B)\phi_j \quad . \tag{40}$$

Thus, comparing Equations 38 and 40,

$$\alpha_{j,n} = \frac{\bar{\bar{\phi}}_j^T[p_jMu_n + Mv_n + Bu_n]}{\bar{\bar{\phi}}_j^T(2p_jM + B)\phi_j} \quad . \tag{41}$$

A subtle point is the fact that the same factor $\alpha_{j,n}$ may be used for both u_n and v_n as indicated in Equation 37. The reasons are that it is valid, under most circumstances, to regard the union of u_n and v_n as an independent vector (x_n), and that the dimension of the vector space for x_n is equal to the number of eigenvalues. Thus the independent eigenvectors $\left\{\begin{array}{c} p_j\phi_j \\ \hline \phi_j \end{array}\right\}$ form a basis for the vector space of x_n except for the circumstance described below.

THE INVERSE POWER METHOD WITH SHIFTS

When the damping matrix, [B], is null, the orthogonality condition, Equation 30, is identical in form to the orthogonality condition for real eigenvalues, $\bar{\bar{\phi}}_j^T M\phi_i = 0$. As a result, considering only the bottom half of Equation 37, the factors $\alpha_{j,n}$ may be calculated from

$$\alpha_{j,n} = \frac{\bar{\bar{\phi}}_j^T M u_n}{\bar{\bar{\phi}}_j^T M \phi_j} \quad , \tag{42}$$

as well as from Equation 41. Since v_n is not rigidly dependent on u_n, Equations 41 and 42 will, in general, give different values of $\alpha_{j,n}$. Thus, if $\boxed{B}$ is null, the assumption that $\alpha_{j,n}$ may be used for both u_n and v_n is not valid. For this case Equation 42 is used to evaluate $(u_n)_p$; $(v_n)_p$ is obtained by substitution of $(u_n)_p$ into the basic iteration algorithm, Equation 14,

$$(v_n)_p = \lambda_o (u_n)_p + \frac{1}{C_n}(u_{n-1})_p \tag{43}$$

The left eigenvector $\bar{\bar{\phi}}_j$ required in Equations 41 and 42 is evaluated after the corresponding right eigenvector, ϕ_j, and the eigenvalue, p_j, have been found. The procedure is to form the lower and upper triangular factors $[L_\ell]$ and $[U_\ell]$ of the dynamic matrix at the accepted value of the eigenvalue,

$$[L_\ell][U_\ell] = [M\, p_j^2 + B\, p_j + K] \quad , \tag{44}$$

and then to solve the equation

$$[M^T p_j^2 + B^T p_j + K]\{\bar{\bar{\phi}}_j\} \equiv [U_\ell]^T[L_\ell]^T\{\bar{\bar{\phi}}_j\} = \{F\} \quad , \tag{45}$$

for $\{\bar{\bar{\phi}}_j\}$ by forward and backward substitution (see Section 2.3). The excitation vector, {F}, may be specified arbitrarily since $[U_\ell]$ is nearly singular (see Section 10.3.8).

The above procedure is similar to that used in finding eigenvectors by the determinant method (see Section 10.3). It has the disadvantage that it requires an additional triangular decomposition of the dynamic matrix for each eigenvalue extracted in order to evaluate the left eigenvector. It is, however, more certain to obtain the correct left eigenvector than any other method that has been examined, and it is relatively efficient for problems with narrow bandwidths.

10.4.4.5 Convergence and Shift Criteria

The convergence and shift criteria for real eigenvalue analysis developed in Sections 10.4.2.2 and 10.4.2.3, and summarized in Equations 9 to 24 of Section 10.4.3, are used with slight modification in complex eigenvalue analysis. It should be noted that the ambiguity with respect to the sign of α_n noted in Equation 24 of Section 10.4.2.2 is still present and that it is removed by comparing the phases of selected components of u_n and u_{n-1}.

The derivation of formulas for the convergence parameters, Λ_1, Λ_2, and δ_n, uses the mass orthogonality property of eigenvectors. Since the mass orthogonality property, $\phi_j{}^T M \phi_i = 0$ for $i \neq j$, does not apply in complex eigenvalue analysis, these formulas require revision.

The orthogonality condition for complex eigenvalue analysis, as derived in Section 10.4.4.2, is

$$\bar{\bar{\phi}}_j{}^T[(p_i + p_j)M + B]\phi_i = 0 \ , \quad i \neq j \ . \tag{46}$$

It cannot be used directly in the convergence criteria because we have no advance knowledge of the left eigenvectors, $\bar{\bar{\phi}}_j{}^T$. For the special case of symmetric matrices, however, the left eigenvectors are equal to the right eigenvectors. For use in the general case, define

$$\delta_{ij} = \phi_i^T\left(M + \frac{1}{p_{1,n-1} + p_{2,n-1}} B\right)\phi_j \ , \tag{47}$$

which is assumed to be normalized to unity for i = j. The parameters $p_{1,n-1}$ and $p_{2,n-1}$ are the current estimates of p_1 and p_2. Also define

$$A_{m,n} = u_m^T\left(M + \frac{1}{p_{1,n-1} + p_{2,n-1}} B\right)u_n \ . \tag{48}$$

Then, substituting for u_m and u_n from Equations 19 and 20 on Page 10.4-5, and using Equation 47, we find that

$$A_{n,n} = \alpha_n^2 \left(1 + \delta_n + \sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) \ , \tag{49}$$

and

$$A_{n-1,n-1} = \alpha_{n-1}^2\left(1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n + \left(\frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) . \tag{50}$$

Substituting from Equations 49 and 50 into Equation 36, a good approximation to the lowest eigenvalue is

$$\Lambda_1 = \frac{1}{C_n}\left(\frac{A_{n-1,n-1}}{A_{n,n}}\right)^{1/2}\left(\frac{1 + \delta_n + \sqrt{\delta_n}\,(\delta_{12} + \delta_{21})}{1 + \left(\frac{\Lambda_2}{\Lambda_1}\right)^2 \delta_n + \frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})}\right)^{1/2} \quad . \tag{51}$$

The error in replacing the last bracketed term by unity is seen to be, since $\delta_n << \sqrt{\delta_n}$, of the order of $\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})$.

The normalized difference of successive trial vectors is, by comparison with Equation 28 of Section 10.4.2.2,

$$\delta u_n = \frac{u_n}{(A_{n,n})^{1/2}} - \frac{u_{n-1}}{(A_{n-1,n-1})^{1/2}} \quad . \tag{52}$$

Substituting from Equations 49 and 50, and assuming $\delta_n << \sqrt{\delta_n}\,(\delta_{12} + \delta_{21}) << 1$,

$$\delta u_n \simeq \frac{u_n}{\alpha_n}\left[1 - \frac{1}{2}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right] - \frac{u_{n-1}}{\alpha_{n-1}}\left[1 - \frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right] \quad , \tag{53}$$

and, using Equations 19 and 20 of Section 10.4.2.2

$$\begin{aligned}\delta u_n &= \left(\phi_1 + \sqrt{\delta_n}\,\phi_2\right)\left(1 - \frac{1}{2}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) \\ &\quad - \left(\phi_1 + \frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,\phi_2\right)\left(1 - \frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12} + \delta_{21})\right) \\ &\simeq \left(1 - \frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\left(\phi_2 - \frac{1}{2}\,(\delta_{12} + \delta_{21})\phi_1\right) \quad .\end{aligned} \tag{54}$$

Similarly

$$\begin{aligned}
\delta u_{n-1} &\simeq \frac{u_{n-1}}{\alpha_{n-1}}\left[1-\frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right]-\frac{u_{n-2}}{\alpha_{n-2}}\left[1-\frac{1}{2}\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right] \\
&\simeq \left(\phi_1+\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,\phi_2\right)\left(1-\frac{1}{2}\frac{\Lambda_2}{\Lambda_1}\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right) \\
&\quad -\left(\phi_1+\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,\phi_2\right)\left(1-\frac{1}{2}\left(\frac{\Lambda_2}{\Lambda_1}\right)^2\sqrt{\delta_n}\,(\delta_{12}+\delta_{21})\right) \\
&\simeq \frac{\Lambda_2}{\Lambda_1}\left(1-\frac{\Lambda_2}{\Lambda_1}\right)\sqrt{\delta_n}\left(\phi_2-\frac{1}{2}(\delta_{12}+\delta_{21})\phi_1\right)
\end{aligned} \tag{55}$$

Therefore, using Equation 47,

$$\begin{aligned}
\Delta_{n,n} &= \delta u_n^T\left(M+\frac{1}{p_{1,n-1}+p_{2,n-1}}B\right)\delta u_n \\
&= \left(1-\frac{\Lambda_2}{\Lambda_1}\right)^2\delta_n\left[1+\frac{1}{4}(\delta_{12}+\delta_{21})^2-\frac{1}{2}(\delta_{12}+\delta_{21})^2\right],
\end{aligned} \tag{56}$$

and

$$\begin{aligned}
\Delta_{n,n-1} &= \delta u_n^T\left(M+\frac{1}{p_{1,n-1}+p_{2,n-1}}B\right)\delta u_{n-1} \\
&= \left(\frac{\Lambda_2}{\Lambda_1}\right)\left(1-\frac{\Lambda_2}{\Lambda_1}\right)^2\delta_n\left[1+\frac{1}{4}(\delta_{12}+\delta_{21})^2-\frac{1}{2}(\delta_{12}+\delta_{21})^2\right] \quad .
\end{aligned} \tag{57}$$

Dividing Equation 56 by Equation 57, it is seen that

$$\frac{\Lambda_2}{\Lambda_1}=\frac{\Delta_{n,n-1}}{\Delta_{n,n}} \quad . \tag{58}$$

From Equation 56,

$$\delta_n=\frac{\Delta_{n,n}}{\left(1-\frac{\Lambda_2}{\Lambda_1}\right)^2\left(1-\frac{1}{4}(\delta_{12}+\delta_{21})^2\right)} \quad . \tag{59}$$

Since we do not know what δ_{12} and δ_{21} are, we will assume that they are zero. It is seen that the resulting approximation to δ_n is accurate only if $(\delta_{12} + \delta_{21})/2 << 1$. If this condition is not satisfied, δ_n, which is used to estimate the degree of convergence, may have a large error. Convergence itself, however, is guaranteed by Equation 36. The only dangers are that the algorithm will stop iterating before satisfactory convergence is achieved, or that it will iterate too long.

10.4.4.6 Search Procedures

In extracting complex eigenvalues, rectangular search regions are set up in the complex plane as shown in Figure 4. It is intended that all eigenvalues within the search regions be extracted (as limited by the desired maximum number of roots). There may be any number of search regions, and the search regions may overlap. The user establishes search regions by means of the coordinates of end points (A_j, B_j) and the width of the region (ℓ_j). The user estimates the number of roots (N_j) in each search region. For problems with real coefficients, all roots are either real or they occur in conjugate complex pairs, and it is inefficient for the user to specify regions in the lower half of the complex plane since the existence of conjugate pairs is assumed. The user will not, however, be prevented from specifying a search in the lower half of the complex plane since it is useful, in many cases, for a search region to include the origin or a band along the real axis.

The tasks performed by NASTRAN in the search procedure are as follows:

1. Divide region I into square subregions and place a starting point at the center of each subregion. The side length of the square subregion is set equal to ℓ_1, and the number of subregions is selected to be

$$\left|\frac{B_1 - A_1}{\ell_1}\right| \le N_r < \left|\frac{B_1 - A_1}{\ell_1}\right| + 1 \quad . \qquad (59)$$

   The sum of the subregions redefines the search region as shown in Figure 5. The redefined search region is symmetrical with respect to the center of the original search region.

2. Select the starting point that is closest to the origin as the first starting point. Select the next closest starting point as the second starting point, etc.

3. Terminate search from any starting point when a root is found outside of a circle passing through the corners of its square, according to rules set forth in Section 10.4.2.5.

4. When the search in region 1 has been completed, proceed to regions 2, 3, etc., in order. Sweep out all previously extracted eigenvectors from whatever region.

10.4.5 Summary of Procedures for Complex Eigenvalue Analysis

The procedures for complex eigenvalue analysis are very similar, and in many instances identical, to those for real eigenvalue analysis. The only major change in the flow diagrams for real eigenvalue analysis (Figures 1, 2, and 3) is the amendment of Figure 2 to include the calculation of the left eigenvector after passage of the convergence tests. Changes in procedure are indicated below according to the numbered blocks in the flow diagrams.

1. Compute Distribution of Starting Points

See Section 10.4.4.6.

2. Select a Starting Point

See Section 10.4.4.6.

4.1 Decompose Dynamic Matrix

The matrix to be decomposed is

$$[\lambda_o^2 M + \lambda_o B + K] \quad , \tag{1}$$

which is, in general, complex and nonsymmetric and where λ_o is either a starting point, λ_s, a moved starting point, λ_s', or a shifted shift point, λ_o'. Double-precision arithmetic with partial pivoting is used.

4.2 One Vector Iteration (see Sections 10.4.4.1 and 10.4.4.4)

The iteration algorithm is, for $B \neq 0$,

$$[\lambda_o^2 M + \lambda_o B + K]\{w_n\} = -[B + \lambda_o M]\{u_{n-1}\} - [M]\{v_{n-1}\} \quad , \tag{2}$$

$$\{\overline{u}_n\} = \frac{1}{C_n}\{w_n\} \quad , \tag{3}$$

$$\{\overline{v}_n\} = \lambda_o\{\overline{u}_n\} + \frac{1}{C_n}\{u_{n-1}\} \quad , \tag{4}$$

$$\{u_n\} = \{\overline{u}_n\} - \sum_j \alpha_j \{\phi_j\} \quad , \tag{5}$$

$$\{v_n\} = \{\overline{v}_n\} - \sum_j \alpha_j p_j \{\phi_j\} \quad , \tag{6}$$

where

$$\alpha_j = \frac{\{\overline{\phi}_j\}^T \{p_j M \overline{u}_n + M \overline{v}_n + B \overline{u}_n\}}{\{\overline{\phi}_j\}^T [2p_j M + B] \{\phi_j\}} \tag{7}$$

C_n = largest element (in magnitude) of $\{w_n\}$,

$\{\phi_j\}$ = previously found eigenvector ,

$\{\overline{\phi}_j\}$ = previously found left eigenvector ,

P_j = previously found eigenvalue ,

λ_o = shift point.

The sweeping operation, Equations 5 and 6, is also applied to starting vectors. The sum on (j) extends over all previously found eigenvalues.

For the special case, [B] = 0, Equation 6 is replaced by

$$\{v_n\} = \lambda_o \{u_n\} + \frac{1}{C_n} \{u_{n-1}\} \quad , \tag{8}$$

and Equation 7 is replaced by

$$\alpha_j = \frac{\{\overline{\phi}_j\}^T [M] \{\overline{u}_n\}}{\{\overline{\phi}_j\}^T [M] \{\phi_j\}} \quad . \tag{9}$$

4.3 Convergence Tests

The convergence tests are identical to those for real eigenvalue analysis except that $\{F_n\}$ is formed in the following manner

$$\{F_n\} = \left[M + \frac{1}{2\lambda_0 + \Lambda_{1,n-1} + \Lambda_{2,n-1}} B\right]\{u_n\} \tag{9a}$$

where $\Lambda_{1,n-1}$ and $\Lambda_{2,n-1}$ are the estimates of Λ_1 and Λ_2 from the previous iteration.

4.8 Compute Left Eigenvector

Left eigenvectors satisfy the equation

$$[p_j^2M^T + p_jB^T + K^T]\{\bar{\phi}_j\} = 0 \quad , \tag{10}$$

where p_j is identically equal to the eigenvalue for the given problem, i.e.,

$$[p_j^2M + p_jB + K]\{\phi_j\} = 0 \quad . \tag{11}$$

See Section 10.4.4.4 for the method employed in calculating $\{\bar{\phi}_j\}$.

5. Have We Found All the Eigenvalues?

The answer is "yes" if the number of eigenvalues found equals the total number estimated to be in the problem. The number found should include conjugate eigenvalues for problems with real M, B, and K matrices. The total number estimated to be in the problem is

$$N_t = 2N_{eq} - N_{oM} \quad , \tag{12}$$

for problems in which $[B] \neq 0$, and

$$N_t = 2N_{eq} - 2N_{oM} \quad , \tag{13}$$

for problems in which $[B] = 0$ and where

N_{eq} = number of rows in dynamic matrix

N_{oM} = number of columns of the mass matrix that are null.

6. Is Eigenvalue Outside Range? (see Figure 5)

The test is:

$$|P_j - \lambda_s| > R_c \quad : \text{ Outside Range,} \tag{14}$$

700

where R_c is the distance from the starting point λ_s to the corners of the starting point's search region, see Figure 5.

7. Select Next-to-Last Trial Vector as Starting Vector

   a. If there was no shift in finding the previous eigenvalue

$$\bar{u}_{i+1,o} = u_{i,N-1} \quad , \tag{15}$$

$$\bar{v}_{i+1,o} = v_{i,N-1} \quad . \tag{16}$$

   b. If there were one or more shifts: Select $\bar{u}_{i+1,o}$ and $\bar{v}_{i+1,o}$ equal to the last vector before the first shift.

8. Too Close to Starting Point? (see Section 10.4.2.4)

The criterion is

$$\varepsilon \left| \frac{R_c}{P_i - \lambda_s} \right| > \varepsilon_3 \ : \ \text{Too close} \tag{17}$$

where

ε = criterion specified by user for convergence of eigenvectors (Default value = 10^{-4})

P_i = eigenvalue just found

λ_s = starting point

R_c = range.

The value stored in the program for ε_3 is .05.

THE INVERSE POWER METHOD WITH SHIFTS

REFERENCE

1. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

702

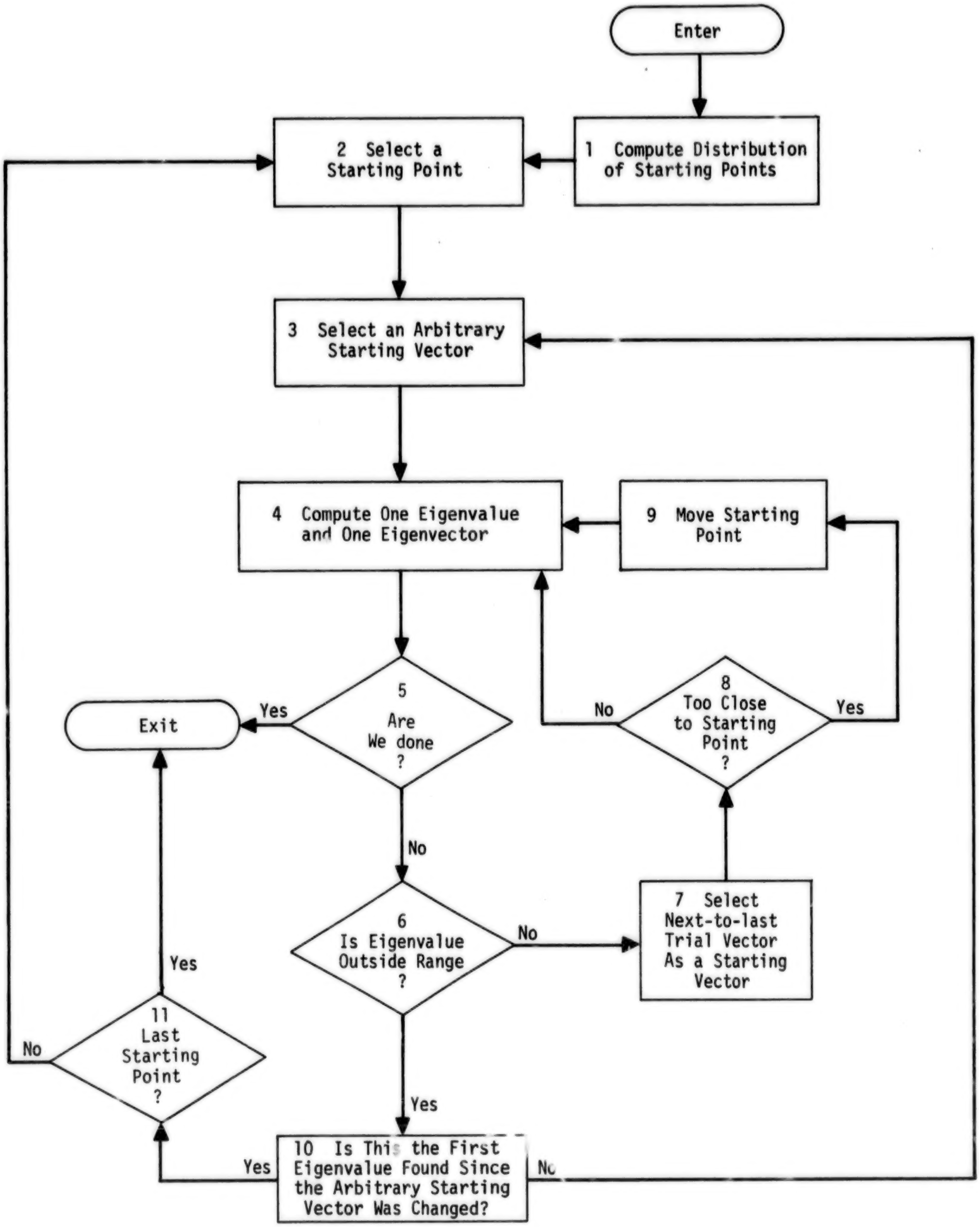

Enter
1 Compute Distribution of Starting Points
2 Select a Starting Point
3 Select an Arbitrary Starting Vector
4 Compute One Eigenvalue and One Eigenvector
9 Move Starting Point
5 Are We done ?
Yes
Exit
No
8 Too Close to Starting Point ?
No
Yes
6 Is Eigenvalue Outside Range ?
No
7 Select Next-to-last Trial Vector As a Starting Vector
Yes
11 Last Starting Point ?
No
Yes
Yes
10 Is This the First Eigenvalue Found Since the Arbitrary Starting Vector Was Changed?
No

Figure 1. Overall flow diagram for inverse power method with shifts.

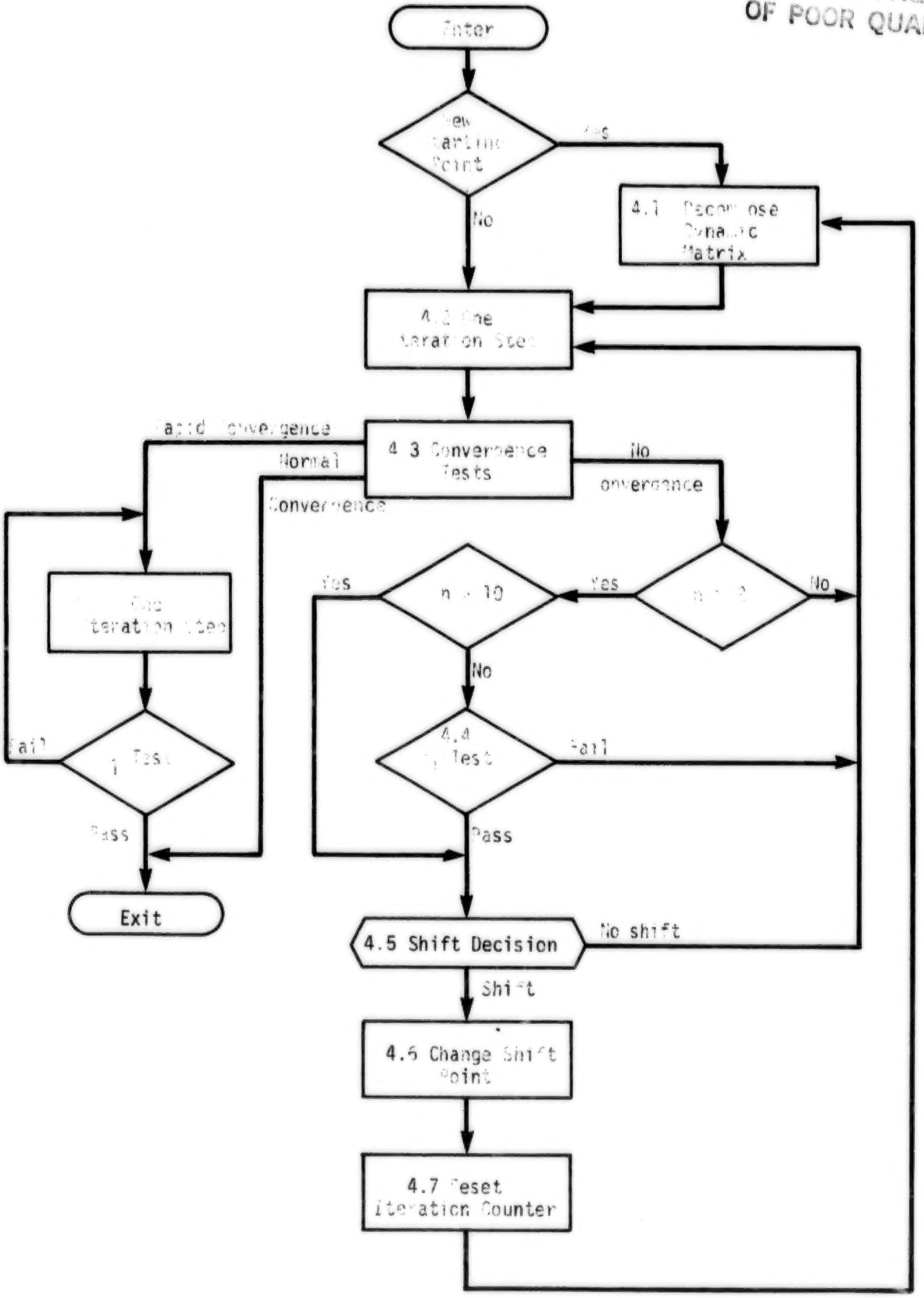

Enter
New Starting Point
Yes
No
4.1 Decompose Dynamic Matrix
4.2 One Iteration Step
4.3 Convergence Tests
Rapid Convergence
Normal Convergence
No Convergence
One Iteration Step
λ_1 Test
Fail
Pass
Exit
n > 10
Yes
No
n > 2
Yes
No
4.4 λ_1 Test
Fail
Pass
4.5 Shift Decision
No shift
Shift
4.6 Change Shift Point
4.7 Reset Iteration Counter

Figure 2. Flow diagram for block 4, compute one eigenvalue and one eigenvector.

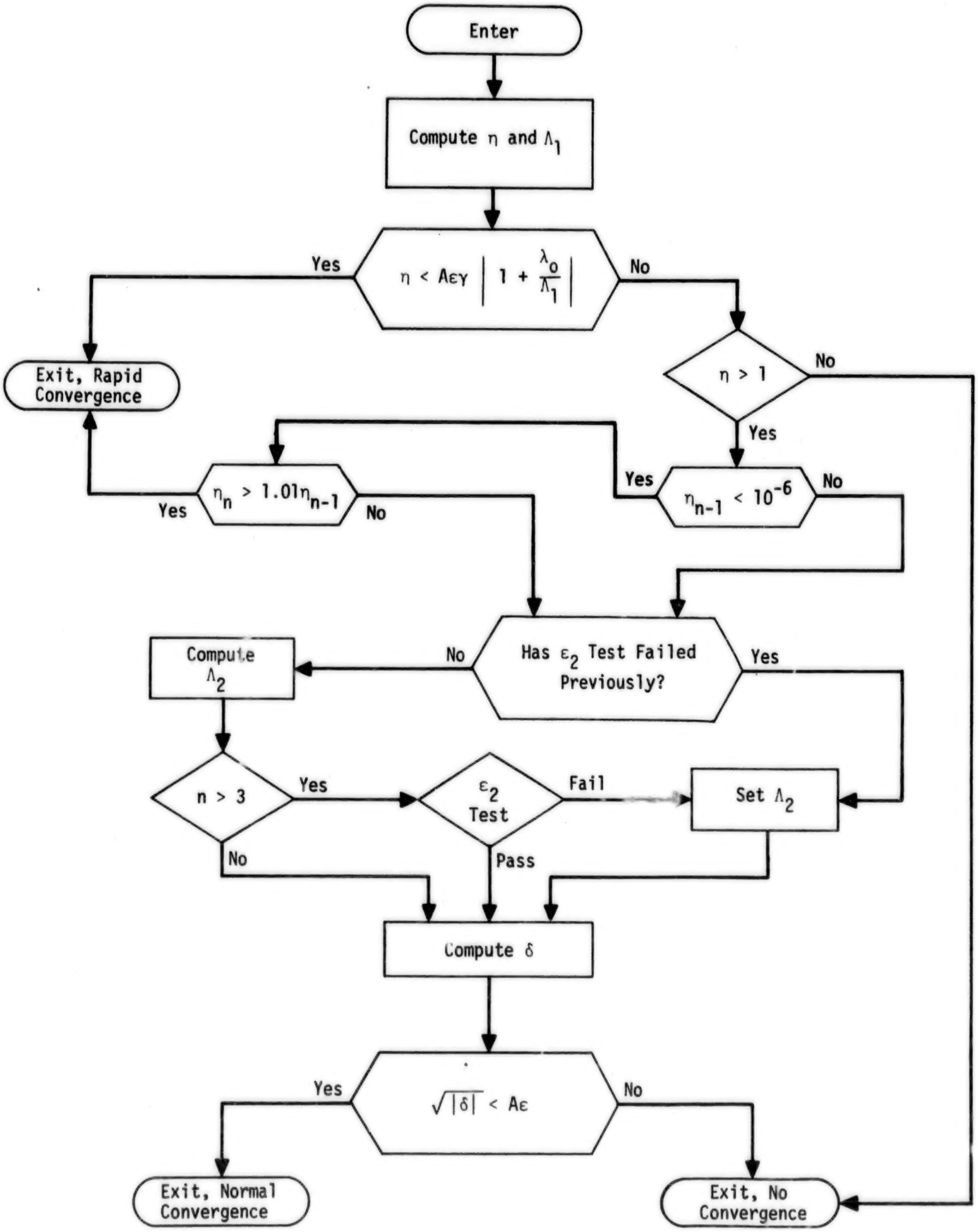

Enter
Compute η and Λ_1
$\eta < A\varepsilon\gamma \left| 1 + \frac{\lambda_0}{\Lambda_1} \right|$
Yes
No
Exit, Rapid Convergence
η > 1
No
Yes
$\eta_{n-1} < 10^{-6}$
Yes
No
$\eta_n > 1.01\eta_{n-1}$
Yes
No
Has ε_2 Test Failed Previously?
No
Yes
Compute Λ_2
n > 3
Yes
No
ε_2 Test
Fail
Pass
Set Λ_2
Compute δ
$\sqrt{|\delta|} < A\varepsilon$
Yes
No
Exit, Normal Convergence
Exit, No Convergence

Figure 3. Flow diagram for block 4.3, convergence tests.

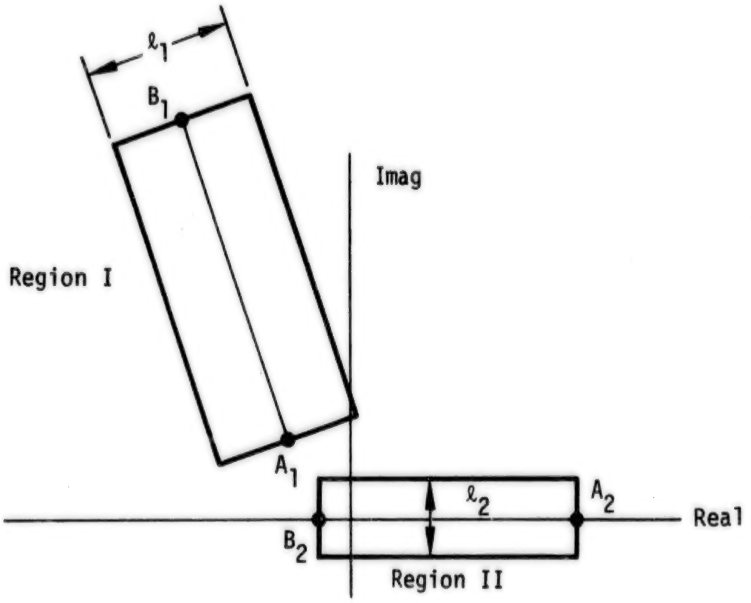

ℓ_1
B_1
Imag
Region I
A_1
ℓ_2
A_2
Real
B_2
Region II

Figure 4. Search regions.

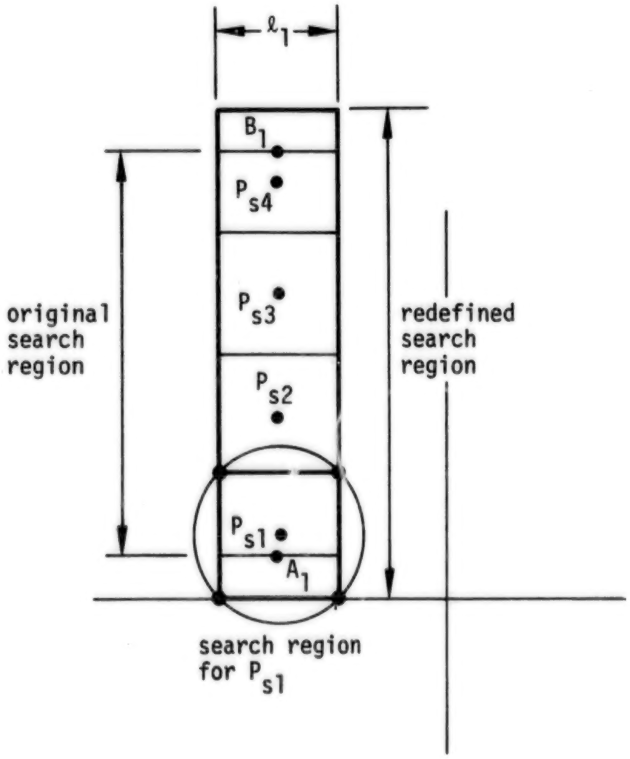

ℓ_1
B_1
P_{s4}
P_{s3}
P_{s2}
P_{s1}
A_1
original search region
redefined search region
search region for P_{s1}

Figure 5. Distribution of starting points within a search region.

10.5 THE UPPER HESSENBERG METHOD

The upper Hessenberg method can be used for extraction of eigenvalues and eigenvectors from general, real or complex matrices. The techniques are more general than those employed in the Tridiagonal Method for Real Symmetric Matrices (Section 10.2). The fundamental reference is Wilkinson (Reference 1). The algorithms of Wilkinson for complex matrices have been automated by Funderlic and Rinzel in the subroutine ALLMAT (see Reference 2), which has been modified for NASTRAN.

The following outline is presented in the logical order of the calculations: Reduction to Canonical Form, Reduction to Upper Hessenberg Form, the QR Iteration, Convergence Criteria, Shifting, Deflation, and Eigenvectors.

10.5.1 Reduction to Canonical Form

The Upper Hessenberg Method requires that the eigenvalue problem be put in canonical form

$$[A - \lambda I]\{\phi\} = 0 \quad . \tag{1}$$

Matrix A is then reduced to the upper Hessenberg form through transformation techniques. The first step is the reduction to the form of Equation 1, which is performed automatically in module CEAD. Two cases are considered

1. $[Mp^2 + Bp + K]\{U\} = 0$ (The general form) (2)

$$A = \left[\begin{array}{c|c} 0 & I \\ \hline -M^{-1}K & -M^{-1}B \end{array}\right]$$

$p = \lambda$

U = The upper half of ϕ. The lower half of ϕ contains the velocity vector and is discarded.

2. $[Mp^2 + K]\{U\} = 0$ (B is not present) (3)

$A = [-M^{-1}K]$

$p = \sqrt{\lambda}$ with $Im(p) > 0$

$U = \phi$

PAGE 707 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

708

In Case 2, the decomposition of M is bypassed if it is an identity matrix. The reduction to canonical form requires that the matrix M must be nonsingular. The order of the problem is doubled when the matrix B is not null.

10.5.2 Reduction to Upper Hessenberg Form

If the given matrix is denoted as [A], it can be reduced to the upper Hessenberg matrix $[A_0]$ by using elementary stabilized transformations. The basic algorithm and two alternatives are given in Reference 1 (pp. 354-355). The selected method uses Equation 9.1 of Reference 1 (p. 355) in its reduction and it appears to be appropriate for the task. The total number of multiplications in the complete reduction is approximately $(5/6)n^3$, which is half the number in Householder's reduction and one-quarter the number in Givens' reduction.

10.5.3 The QR Iteration

The QR iteration of Francis (Reference 3) is defined by the relations (Reference 1, p. 515)

$$[A_0^{(s)}] = [Q^{(s)}][R^{(s)}] \quad , \tag{4}$$

$$[A_0^{(s+1)}] = [R^{(s)}][Q^{(s)}] \quad , \tag{5}$$

where $[Q^{(s)}]$ is the product of the (n-1) elementary unitary transformations necessary to reduce $[A_0^{(s)}]$ to the upper triangular form $[R^{(s)}]$ with positive real diagonal elements,

$$[Q^{(s)}] = [T_s^{(1)}][T_s^{(2)}] \ \dots \ [T_s^{(n-2)}][T_s^{(n-1)}] \quad , \tag{6}$$

so that

$$[R^{(s)}] = [Q^{(s)}]^{-1}[A^{(s)}] \quad . \tag{7}$$

The transformation matrices $[T_s^{(j)}]$ are the Givens' rotations as discussed in Reference 1 (p. 239-240) (and in this manual, Section 10.2, in real form for real matrices) but in complex form for complex matrices. The iteration is continued until the n^th^ diagonal element $\left|a_{n,n-1}^{(s)}\right| < \varepsilon$, the convergence test, at which point the smallest eigenvalue $\lambda_1 = a_{n,n}^{(s)}$; if the convergence proceeds so that $\left|a_{n-1,n-2}^{(s)}\right| < \varepsilon$ before $\left|a_{n,n-1}^{(s)}\right| < \varepsilon$, the two smallest eigenvalues are the roots of

$$\begin{vmatrix} a_{n-1,n-1}^{(s)} - \lambda & a_{n-1,n}^{(s)} \\ a_{n,n-1}^{(s)} & a_{n,n}^{(s)} - \lambda \end{vmatrix} = 0 \quad . \tag{8}$$

The roots will be complex for complex matrices, and either real or complex conjugates for real matrices. Before each iteration, the subdiagonal elements should be tested and if some $|a_{i,i-1}^{(s)}| < \varepsilon$, the matrix should be split according to this occurrence, and the iteration continued with the lower main submatrix only.

10.5.4 Convergence Criteria

The convergence criteria suggested by Wilkinson (Reference 1, p. 526) is based on the Euclidean norm of the matrix $||A_0||_E$ and is

$$\varepsilon = 2^{-t} \, ||A_0||_E \quad , \tag{9}$$

for floating-point calculations with mantissas of t binary bits. The Euclidean norm (Reference 1, p. 57) is found from

$$||A||_E^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \quad . \tag{10}$$

Decimal equivalents of Equation 9 are used.

10.5.5 Shifting

Since the QR iteration converges to the smallest eigenvalue, the convergence can be accelerated by shifting, i.e., by subtracting scalar matrices from the original matrix. The matrix $[A_0^{(s)}]$ is replaced by the difference $[A_0^{(s)}] - k_s[I]$ after each iteration, in which k_s is an estimate of the eigenvalue. The shift eigenvalue k_s is that root of Equation 8, p_s or q_s, that makes $|a_{n,n}^{(s)} - p_s|$ or $|a_{n,n}^{(s)} - q_s|$ a minimum. The shifted algorithm then becomes (Reference 1, p. 524)

$$[A_0^{(s)}] - k_s[I] = [Q^{(s)}][R^{(s)}] \quad , \tag{11}$$

and

$$[A_0^{(s+1)}] = k_s[I] + [R^{(s)}][Q^{(s)}] \quad . \tag{12}$$

Equations 11 and 12 represent the algorithm for a shift to a single eigenvalue and is appropriate for a complex matrix or a real matrix with all real eigenvalues.

10.5.6 Deflation

When convergence to a single eigenvalue occurs, i.e., when $|a_{n,n-1}^{(s)}| < \varepsilon$, the Hessenberg matrix $[A_0]$ is deflated by elimination of its last row and column and the principal submatrix $[A_1]$ of order one less is the Hessenberg form for seeking the next eigenvalue. (Note: the subscript on A denotes the number of eigenvalues removed from $[A_0]$.) If convergence occurs to a pair of eigenvalues, i.e., $|a_{n-1,n-2}^{(s)}| < \varepsilon$, the matrix $[A_0]$ is deflated by deleting the last two rows and columns, and the principal submatrix $[A_2]$ of order two less becomes the basis for seeking the next eigenvalue. Each deflation removes either one or two eigenvalues depending on the two convergence tests.

10.5.7 Eigenvectors

The inverse power method with shifts (Reference 1, pp 626-628) converges rapidly to the eigenvector corresponding to each shift eigenvalue. This algorithm for real and complex matrices has been discussed thoroughly in Section 10.4. No further discussion is required here.

THE UPPER HESSENBERG METHOD

REFERENCES

1. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

2. Funderlic, R. E., and Rinzel, J., "ALLMAT, A FORTRAN IV Arbitrary Matrix Eigensystem Solver," SHARE Program Library, SDA No. 3441, 2 March 1966.

3. Francis, J. G. F., "The QR Transformation - a Unitary Analogue to the LR Transformation," Parts 1 and 2, The Computer Journal, Vol. 4, No. 3 (October 1961) and No. 4 (January 1962).

10.6 THE TRIDIAGONAL REDUCTION (FEER) METHOD

10.6.1 Introduction

The Tridiagonal Reduction or FEER Method is an automatic matrix reduction scheme whereby the eigensolutions in the neighborhood of a specified point in the eigenspectrum can be accurately extracted from a tridiagonal eigenvalue problem whose order is much lower than that of the full problem. Specifically, the order, m, of the reduced problem is never greater than

$$m = 2\bar{q} + 10, \tag{1}$$

where $\bar{q}$ is the desired number of accurately computed eigenvalues. Thus, the intrinsic power of the method lies in the fact that the size of the reduced eigenvalue problem is of the same order of magnitude as the number of desired roots, even though the discretized system model may possess thousands of degrees of freedom. The process is effected without arbitrary lumping of masses or other physical quantities at selected node points and thus avoids one of the basic weaknesses of the Guyan Reduction Method (Reference 1) and other techniques (References 2 and 3) requiring a judicious selection of the degrees of freedom to be retained.

Tridiagonal reduction was first suggested by Crandall (Reference 4) as a truncated version of the Lanczos Algorithm (Reference 5). However, it was soon discovered that the original scheme possessed numerical instabilities (References 6 and 7). The necessary improvements to correct these weaknesses were made by Ojalvo and Newman (Reference 8) who were the first to develop a successful tridiagonal reduction program for large scale structural vibration problems. Further refinements were later introduced by Newman and Pipano in the FEER computer program (References 9 and 10), including the following extended features:

1. Highly efficient numerical computation schemes which take advantage of matrix banding and sparsity.

2. Calculation of upper and lower error bounds on the extracted eigenvalue estimates.

3. Accommodation of singular mass matrices and stiffness matrix singularities associated with rigid body modes.

PAGE 713 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

C-8

The last capability, which is also present in the Inverse Power Method with Shifts, overcomes a basic restriction of the Tridiagonal (Givens) and upper Hessenberg methods, namely, the need to eliminate massless degrees of freedom.

For real eigenvalue analysis, the computational speed of the Tridiagonal Reduction Method is, in many cases, almost as fast as the Givens (Reference 11) and Householder methods (Reference 12) when all the eigensolutions are calculated, and becomes increasingly more efficient as the number of required eigensolutions is reduced. In addition, in order to avoid prohibitively long running times, both the Givens and Householder methods require the use of a relatively large computer central memory for even moderate problem sizes, while the Tridiagonal Reduction Method is extremely efficient with regard to core requirements.

In the case of complex eigenvalue analysis, the above remarks are also applicable for the complex version of FEER when compared to the Upper Hessenberg method.

As shown in the ensuing development, the Tridiagonal Reduction Method employs only a single initial shift of eigenvalues and hence usually requires only one matrix decomposition. It consequently tends to be much more efficient than the Inverse Power Method when more than one or two eigensolutions are required.

10.6.2 Theory for Real Eigenvalue Analysis

The development of the Tridiagonal Reduction Method and its implementation for real eigenvalue analysis, as typified by structural vibration and buckling problems, are described in the following subsections. The restrictions on the use of the method are as follows:

1. For structural vibration mode applications, the method extracts a preselected number of eigenvalues which are closest to a specified shift value, λ_o, rather than computing the eigenvalues in a prescribed range.

2. In buckling problems, a preselected number of eigenvalues of smallest magnitude are obtained, i.e., no shifting is performed. Physically, this implies that the buckling load parameters, whether positive or negative, are computed in order of increasing magnitude.

The basic steps employed in the method are as follows. First, the original eigenvalue problem

$$[K - \lambda_a M]\{\phi\} = 0, \tag{2}$$

is converted to a symmetric inverse form

$$[B]\{X\} = \Lambda[D]\{X\}, \tag{3}$$

where

$$\Lambda = \frac{1}{\lambda_a - \lambda_o} \tag{4}$$

and λ_o is a shift value which is used in structural vibration mode applications. Second, the tridiagonal reduction algorithm is employed to transform Equation 3 into a tridiagonal form of reduced order. Third, the eigenvalues of the reduced matrix are extracted using a Q-R algorithm similar to that described in Section 10.2.3. Fourth, the corresponding eigenvectors are computed and converted to physical form. Finally, upper and lower bounds on the extracted eigenvalues are obtained.

A detailed summary of the computational procedures and explanatory flow diagrams are provided in Section 10.6.3.

10.6.2.1 Preliminary Operations

The problem is to find a specified number of real eigenvalues and corresponding eigenvectors for

$$[K - \lambda_a M]\{\phi\} = 0. \tag{5}$$

It is further required that these eigensolutions constitute the set lying closest to a specified point, λ_o, in the eigenspectrum.

The definitions of the eigenvalue, λ_a, the matrices [K] and [M], and their mathematical properties, depend on the type of problem being solved within the program environment. For real analysis, only two separate problem types need be considered; structural vibration and buckling problems. The matrix definitions and mathematical distinctions for these two cases are summarized in the following table:

Table 1. Problem Formulations

Problem Type	Quantity	Definition	NASTRAN Notation	Most General Properties
Structural Vibration Modes	[K]	Stiffness Matrix - analysis set	$[K_{aa}]$	Symmetric, non-negative, semidefinite matrix
	[M]	Mass Matrix - analysis set	$[M_{aa}]$	Same as above
	λ_a	Square of a circular natural frequency	ω^2	Positive
Buckling	[K]	Stiffness Matrix - analysis set	$[K_{aa}]$	Symmetric, positive-definite matrix
	[M]	Differential Stiffness Matrix - analysis set	$[K^d_{aa}]$	Symmetric, indefinite matrix
	λ_a	Buckling Load Parameter	$-\lambda$	Positive or negative

The essential mathematical differences between the two types of problems center around the properties of the [M] matrix, which is non-negative for vibration mode problems, but indefinite for buckling problems, thereby permitting the existence of both positive and negative eigenvalues in the latter case. In addition, the stiffness matrix may be singular for vibration problems while it is always positive definite in buckling applications, which implies that the buckling analysis is performed on a kinematically stable structure.

In summary, the two problems under consideration are of the forms

$$[K_{aa} - \omega^2 M_{aa}]\{\phi\} = 0 \qquad \text{(structural vibrations)} \tag{6}$$

and

$$[K_{aa} + \lambda K^d_{aa}]\{\phi\} = 0. \qquad \text{(buckling)} \tag{7}$$

Further, if the user requests vibration modes in the neighborhood of a specified frequency, ω_o, Equation 6 can be written as

$$[\bar{K}]\{\phi\} = \lambda'[M_{aa}]\{\phi\}, \tag{8}$$

where

$$[\bar{K}] = [K_{aa} - \omega_o^2 M_{aa}], \tag{9}$$

and

$$\lambda' = \omega^2 - \omega_o^2. \tag{10}$$

The resulting effective stiffness matrix, $[\bar{K}]$, is indefinite in this case, since it possesses both positive and negative eigenvalues. This requires that a non-square root decomposition scheme be used in subsequent operations. However, $\omega_o = 0$ is taken as a default value, or it may be specified by the user. In this case, a specified number of natural frequencies starting with the lowest will be computed. In order to utilize a more efficient Cholesky decomposition of $[\bar{K}]$ under these conditions, a small negative shift $\lambda_o = -\alpha^2$ (see Section 10.6.2.6) is used, yielding

$$[\bar{K}] = [K_{aa} + \alpha^2 M_{aa}], \tag{11}$$

and

$$\lambda' = \omega^2 + \alpha^2. \tag{12}$$

718

It is easy to prove that the resulting effective stiffness matrix $[\bar{K}]$ is positive-definite provided that the system masses generate positive kinetic energy due to any kinematically admissible rigid body motions of the structure. This requirement is always satisfied by the mass matrix in a physically well posed problem, thereby allowing a Cholesky square-root decomposition to be performed when the roots are computed in the neighborhood of zero. Since no shifting is performed in buckling problems, the effective stiffness matrix is $[\bar{K}] = [K_{aa}]$, which is always positive-definite, again permitting the use of a Cholesky decomposition.

In any event, a decomposition or factoring of $[\bar{K}]$ is next performed:

$$[\bar{K}] = [L][\ulcorner d \lrcorner][L]^T, \quad \text{(shifted vibration mode problems)} \tag{13}$$

or

$$[\bar{K}] = [C][C]^T, \quad \text{(buckling problems or vibration modes in the neighborhood of zero desired)} \tag{14}$$

where $[L]$ and $[C]$ are lower triangular factors and $[\ulcorner d \lrcorner]$ is a diagonal matrix.

To facilitate computation of eigenvalues closest to the point of interest within the eigenspectrum, inverse forms of the eigenvalue problems are employed, as in the Inverse Power Method with Shifts.

The general form of the inverse problem may be written as

$$[B]\{X\} = \Lambda[D]\{X\}, \tag{15}$$

where the above terms are defined as follows:

Table 2. Inverse Eigenproblem Definitions

Problem Type	[B]	[D]	{X}	Λ
1. Shifted Vibration Modes	$[M_{aa}]([L]^{-1})^T[\lceil d \rfloor]^{-1}[L]^{-1}[M_{aa}]$	$[M_{aa}]$	$\{\phi\}$	$\frac{1}{\omega^2 - \omega_o^2}$
2. Unshifted Vibration Modes (in the neighborhood of zero frequency)	$[C]^{-1}[M_{aa}]([C]^{-1})^T$	[I] (Identity Matrix)	$[C]^T\{\phi\}$	$\frac{1}{\omega^2 + \alpha^2}$
3. Buckling Modes	$[C]^{-1}[K_{aa}^d]([C]^{-1})^T$	[I]	$[C]^T\{\phi\}$	$-\frac{1}{\lambda}$

The above triangular matrix inverses are treated as purely operational symbols, since in actual numerical computations vectors defined, for example, by

$$\{a\} = [L]^{-1}\{b\}, \tag{16}$$

$$\{\bar{a}\} = ([L]^{-1})^T\{\bar{b}\} \tag{17}$$

are obtained from the solutions of

$$[L]\{a\} = \{b\} , \tag{18}$$

$$[L]^T\{\bar{a}\} = \{\bar{b}\} , \tag{19}$$

employing forward and backward passes.

10.6.2.2 The Reduction Algorithm

A reduction of the order of the eigenvalue problem, Equation 15, is effected through the transformation

$$\underset{n\times 1}{\{\bar{X}\}} = \underset{n\times m}{[V]}\ \underset{m\times 1}{\{y\}} , \tag{20}$$

where $\{\bar{X}\}$ is an approximation of $\{X\}$, n is the order of the unreduced problem, and $m \le n$. The transformation matrix is taken to be orthonormal to [D]*, so that

$$[V]^T[D][V] = [I]. \tag{21}$$

From Equations 15, 20 and 21 it is seen that

$$[A]\{y\} = \bar{\Lambda}\{y\}, \tag{22}$$

where

$$\underset{m\times m}{[A]} = [V]^T[B][V], \tag{23}$$

and $\bar{\Lambda}$ is an approximation of the eigenvalue, Λ.

Thus, Equation 22 is an m^{th} order eigenvalue problem where $m \leq n$. The value of m is established according to the criteria given in Section 10.6.2.3.

The essence of the reduction scheme lies in the choice of the transformation matrix [V]. In the present case the Lanczos algorithm is used to build up the [V] matrix, vector by vector, i.e.,

$$\underset{n\times m}{[V]} = [\{v_1\},\{v_2\},......,\{v_m\}], \tag{24}$$

such that the reduced mxm matrix [A] is tridiagonal and its eigenvalues accurately approximate the roots of Equation 15 having the largest magnitude (or, equivalently, the roots of the physical model closest to the specified point of interest in the eigenspectrum).

Define the matrix

$$[\bar{B}] = [D]^{-1}[B]^{**}, \tag{25}$$

and let

---

*In problem types 2 and 3 (see Table 2), [D] is the identity matrix and, therefore, $[V]^T[V] = [I]$.

**For shifted vibration mode problems, where $[D] = [M_{aa}]$, this operation is purely symbolic (since $[M_{aa}]$ may be singular) and Equation 25 merely implies that $[B] = ([L]^{-1})^T[\ d\ ]^{-1}[M_{aa}]$.

$$
\begin{aligned}
d_{i+1}\{v_{i+1}\} = \{\bar{v}_{i+1}\} &= [\bar{B}]\{v_i\} - a_{i,i}\{v_i\} - a_{i,i-1}\{v_{i-1}\} \\
&\qquad \ldots\ldots\ldots\ldots\ldots - a_{i,1}\{v_1\} \\
&= [\bar{B}]\{v_i\} - \sum_{j=1}^{i} a_{i,j}\{v_j\}; \quad i = 1, m-1 \qquad (26)
\end{aligned}
$$

where $\{v_1\}$ is a starting vector (see Section 10.6.2.4) and all m $\{v\}$-vectors are orthonormal to [D]. This implies that

$$a_{i,j} = \{v_j\}^T[B]\{v_i\}, \qquad (27)$$

while d_{i+1} is a normalizing factor given by,

$$d_{i+1} = [\{\bar{v}_{i+1}\}^T[D]\{\bar{v}_{i+1}\}]^{1/2}. \qquad (28)$$

The recurrence relationship, Equation 26, when carried out to one additional step (i.e., over the range i = 1, m), can be expressed in the following matrix form:

$$\underset{n\times n}{[\bar{B}]}\underset{n\times m}{[V]} = \underset{n\times m}{[V]}\underset{m\times m}{[H]} + d_{m+1}\underset{n\times m}{[G_{m+1}]}, \qquad (29)$$

where [H] is an upper Hessenberg matrix given by,

$$[H] = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{m1} \\ d_2 & a_{22} & a_{32} & \cdots & a_{m2} \\ & d_3 & a_{33} & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & d_m & a_{mm} \end{bmatrix}, \qquad (30)$$

and

$$[G_{m+1}] = [\{0\} \,\vdots\, \{0\} \,\vdots\, \ldots\ldots \,\vdots\, \{0\} \,\vdots\, \{v_{m+1}\}]. \qquad (31)$$

Premultiplying Equation 29 by $[V]^T[D]$, it can be seen that

$$[V]^T[B][V] = [V]^T[D][V][H] + [0], \tag{32}$$

or

$$[A] = [H]. \tag{33}$$

However, the matrix [A] is symmetric and hence [H] is symmetric, requiring that it be tridiagonal. It follows that

$$a_{i,j} = 0; \quad j < i-1, \tag{34}$$

and

$$d_i = a_{i,i-1}, \tag{35}$$

while Equation 29 assumes the form

$$[\bar{B}][V] = [V][A] + d_{m+1}[G_{m+1}]. \tag{36}$$

As a consequence of Equations 33 to 35, the reduced tridiagonal eigenvalue problem, Equation 22, is

$$[A]\{y\} = \begin{bmatrix} a_{11} & d_2 & & \\ d_2 & a_{22} & d_3 & \\ & d_3 & a_{33} \; \ddots & d_m \\ & & d_m & a_{mm} \end{bmatrix} \{y\} = \bar{\Lambda}\{y\} , \tag{37}$$

and the matrix coefficients are theoretically given by the simplified recurrence formulas

$$\left.\begin{aligned} a_{i,i} &= \{v_i\}^T[B]\{v_i\} \\ \{\bar{v}_{i+1}\} &= [\bar{B}]\{v_i\} - a_{i,i}\{v_i\} - d_i\{v_{i-1}\} \\ d_{i+1} &= [\{\bar{v}_{i+1}\}^T[D]\{\bar{v}_{i+1}\}]^{1/2} , \end{aligned}\right\} \quad i = 1,m \tag{38}$$

$$\{v_{i+1}\} = \frac{1}{d_{i+1}}\{\bar{v}_{i+1}\} ; \qquad i = 1, m - 1 \tag{39}$$

where the sequence is initialized by choosing a random starting vector for $\{v_1\}$ and setting

$d_1 = 0, \{v_0\} = \{0\}$.

Although the final off-diagonal term, d_{m+1}, given by Equations 38 is not needed to construct the tridiagonal reduced matrix, it is calculated and saved for use in establishing error bounds (see Section 10.6.2.7). In addition, the above algorithm is modified in the computer program as follows:

1. Each vector $\{v_{i+1}\}$, calculated by Equation 39 is re-orthogonalized to all previously computed $\{v\}$-vectors and eigenvectors, as described in Section 10.6.2.5, before re-entering Equations 38.

2. The actual size, m, of the reduced eigenproblem is established by the criteria given in Section 10.6.2.3, one of the restrictions being that it cannot exceed the rank of the matrix $[M_{aa}]$ for vibration mode problems or $[K^d_{aa}]$ for buckling problems.

The eigenvalues, $\bar{\Lambda}$, and eigenvectors, $\{y\}$ of Equation 37 are extracted using a Q-R algorithm and eigenvector computational procedure similar to that described in Sections 10.2.3 and 10.2.4. They are then converted to physical form as follows:

$$\bar{\lambda}_i = -\frac{1}{\bar{\Lambda}_i}, \qquad \text{(buckling problems)} \tag{40}$$

$$\bar{\omega}_i^2 = \frac{1}{\bar{\Lambda}_i} - \alpha^2, \qquad \text{(unshifted vibration mode problems)} \tag{41}$$

$$\bar{\omega}_i^2 = \frac{1}{\bar{\Lambda}_i} + \omega_0^2, \qquad \text{(shifted vibration mode problems)} \tag{42}$$

$$\{\bar{\phi}_i\} = ([C]^{-1})^T[V]\{y_i\}, \text{(buckling or unshifted vibration mode problems)} \tag{43}$$

$$\{\bar{\phi}_i\} = [V]\{y_i\}. \qquad \text{(shifted vibration mode problems)} \tag{44}$$

In addition, the vectors $\{y_i\}$ are normalized ($\{y_i\}^T\{y_i\} = 1$) for convenience in establishing error bounds (see Section 10.6.2.7).

10.6.2.3 Criteria for the Size of the Reduced Eigenvalue Problem

The total number of eigensolutions, including any existing rigid body modes, is equal to the rank, r, of the matrix $[M] = [M_{aa}]$ or $[M] = [K^d_{aa}]$, depending on whether a buckling or vibration mode problem is being solved. Thus, the size of the reduced problem, m, cannot be greater than r. If, for example, [M] is diagonal, then the maximum permissible value of m is equal to the total number of non-zero diagonal entries. In addition, if f eigensolutions have previously been computed, these must be swept out of the problem by making all {v} vectors orthogonal to the previously computed eigenvectors. This implies that the maximum size of the reduced problem is further reduced to

$$\bar{r} = r - f. \tag{45}$$

As a result of numerical experiments and application experiences (References 8, 9 and 13), it has been found that in cases where $m \ll \bar{r}$, a first grouping of more than m/2 eigenvalues closest to the shift point are in accurate agreement with the corresponding number of exact eigenvalues, provided that $12 \leq m \leq \bar{r}$. The remaining reduced-system roots are spread across the reamining exact eigenspectrum.

Thus, if the user requests a total of q eigenvalues closest to a specified numerical value*, the order of the reduced problem solved will be

$$m = \min[2\bar{q} + 10, \bar{r}], \tag{46}$$

where

$$\bar{q} = q - f. \tag{47}$$

It should be noted that in all cases $m \leq \bar{r}$, and whenever m is set equal to $\bar{r}$, all the eigensolutions of the unreduced problem, which have not been previously computed, are generated.

From the above discussion it is apparent that, depending on the number of eigensolutions requested, the rank of [M] may have an influence on the order of the reduced problem which is generated by the program. Although the user should never request more than $\bar{r}$ eigensolutions, the value of $\bar{r}$ may not always be a simple matter to calculate, particularly in buckling problems. Therefore, the computational scheme has been designed to resolve the question of rank in the following manner:

---

*q includes the number of previously computed eigensolutions, f. These consist of modes generated prior to a restart plus rigid body modes generated by using a SUP0RT card in the bulk data deck.

1. The matrix [M] is first checked for inordinately small off-diagonal elements, i.e., those which lie in the round-off range of the computer. These terms can sometimes introduce artifically large physical eigenvalues and are therefore eliminated. Any off-diagonal element for which

$$|m_{ij}| \le 10^{-2t/3}\ |m_{ii}|;\ i \ne j,\ m_{ii} \ne 0, \tag{48}$$

   is set equal to zero, where t is the number of decimal digits carried in the computation of the [M] matrix.

2. The number of non-null columns or rows of the above modified [M] matrix is counted and designated as $\bar{n}$. Since the rank, r, cannot be greater than $\bar{n}$, the program initially sets $\bar{r} = \bar{n} - f$ in Equation 46 to provide a tentative size, m, of the reduced eigenvalue problem. If the user has asked for more than $\bar{r}$ eigensolutions, the program will try to find all the existing solutions.

3. If the re-orthogonalization tests (see Section 10.6.2.5) fail for some vector $\{v_{i+1}\}$, this is an indication that a null vector has been generated because the maximum number of r - f linearly independent {v} vectors have already been obtained. The recurrence sequence is then terminated and the order of the reduced eigenproblem is further reduced to m = i.

10.6.2.4 Choice of the Initial Trial Vector and Restart Vectors

Prior to tridiagonal reduction, the original eigenvalue problem

$$[K - \lambda_a M]\{\phi\} = 0, \tag{49}$$

is cast in the inverse form

$$[B]\{X\} = \Lambda[D]\{X\}\ , \tag{50}$$

or

$$[\bar{B}]\{X\} = \Lambda\{X\}, \tag{51}$$

where

$$\Lambda = \frac{1}{\lambda_a - \lambda_o}. \tag{52}$$

From Equation 49 it can be seen that if the problem size is designated as n and r is the rank of the [M] matrix, then there are (n - r) spurious eigensolutions corresponding to $\Lambda = 0$ or, equivalently, $\lambda_a \to \infty$. If, for example, a relatively small number of nodal masses or differential stiffness elements are employed, then $[\bar{B}]$ contains a large multiplicity of zero eigenvalues which are of no interest and cause numerical difficulties. These are eliminated from the reduced tridiagonal problem by employing a constrained sub-set of v-vectors having zero projection on the set of eigenvectors associated with $\Lambda = 0$.

To accomplish this, use is made of the fact that any non-null vector $\{\bar{v}_1\}$, generated from any other non-null vector $\{w\}$, through

$$\{\bar{v}_1\} = [\bar{B}]\{w\} \tag{53}$$

will contain no components of the eigenvectors corresponding to $\Lambda = 0$. This can be seen as follows.

First, express $\{w\}$ as a linear combination of all the eigenvectors of $[\bar{B}]$:

$$\{w\} = \sum_{i=1}^{r} c_i\{X_i\} + \sum_{i=r+1}^{n} c_i\{X_i\}, \tag{54}$$

where $\{X_i\}$, $(i = 1, r)$, are the eigenvectors for $\Lambda \neq 0$ and $\{X_i\}$, $(i = r + 1, n)$, are the eigenvectors for $\Lambda = 0$.

Next, substitute Equation 54 into Equation 53, giving

$$\{\bar{v}_1\} = \sum_{i=1}^{r} c_i[\bar{B}]\{X_i\} + \sum_{i=r+1}^{n} c_i[\bar{B}]\{X_i\}. \tag{55}$$

Since

$$[\bar{B}]\{X_i\} = \Lambda_i\{X_i\}; \quad i = 1, r, \tag{56}$$

and

$$[\bar{B}]\{X_i\} = 0 \quad ; \quad i = r + 1, n, \tag{57}$$

it follows that

$$\{\bar{v}_1\} = \sum_{i=1}^{r} c_i \Lambda_i \{X_i\} . \tag{58}$$

Thus, $\{\bar{v}_1\}$ contains no eigenvectors corresponding to $\Lambda = 0$, and is a null vector only if $\{w\}$ happens to be a $\Lambda = 0$ eigenvector. Further, it is easy to see that after $\{\bar{v}_1\}$ is normalized

$$\{v_1\} = \left[\frac{1}{\{\bar{v}_1\}^T[D]\{\bar{v}_1\}}\right]^{1/2}\{\bar{v}_1\}, \tag{59}$$

the next vector generated,

$$\{\bar{v}_2\} = [\bar{B}]\{v_1\} - a_{1,1}\{v_1\}, \tag{60}$$

as well as all subsequent trial vectors will be free of $\Lambda = 0$ eigenvectors.

Employing a somewhat similar argument, it can be shown that the most desirable initial trial vector, $\{\bar{v}_1\}$, and hence the vector $\{w\}$ from which it is generated, should contain all components of the eigenvectors for $\Lambda \neq 0$. However, since there is generally no a-priori knowledge of the modal matrix, $\{w\}$ should be selected in such a way as to make it as "irregular" as possible with respect to the system of eigenvectors so that it is most likely to contain a mixture of all the mode shapes. It has been found that this is best achieved by using a random or pseudo-random number generator to obtain the elements of $\{w\}$.

If the vector $\{w\}$ selected in the above manner should, by some chance, be deficient in eigenvector components, then a null v-vector may be generated at some point by the recurrence algorithm (Equations 38 and 39). In the context of a finite digit computer, this is indicated by the appearance of an off-diagonal term, d_{i+1}, which is exceedingly small compared to the corresponding diagonal term, $a_{i,i}$, in the reduced tridiagonal matrix. The test used is that

$$|d_{i+1}| \leq 10^{2-t}\,|a_{i,i}| \tag{61}$$

implies that the newly generated vector $\{\bar{v}_{i+1}\}$, is null, where t is the number of decimal digits carried by the computer. In this event, d_{i+1} is set exactly equal to zero and a new restart vector is employed for $\{\bar{v}_{i+1}\}$. This vector is generated exactly as in the case of the initial trial vector $\{v_1\}$, but using a different pseudo-random number seed. The recurrence algorithm for the generation of the v-vectors is subsequently continued in the usual manner until the required number of vectors has been generated.

A further constraint on each v-vector, including the initial one, is that it be orthogonal to all previously generated v-vectors and previously calculated eigenvectors**. The imposition of these additional constraints is discussed in the next subsection.

10.6.2.5 Sweeping-out of Previously Obtained Eigenvectors and Re-orthogonalization of the Trial Vectors

Assume that a combination of f rigid body and non-rigid body eigenvectors has already been extracted prior to the current application of the tridiagonal reduction method. Let these vectors be designated by $\{\tilde{X}_1\}$, $\{\tilde{X}_2\}, \ldots, \{\tilde{X}_f\}$. In order to avoid regenerating these previous eigensolutions, which would be inefficient, the initial trial vector, $\{v_1\}$, obtained in the manner just described, should be made orthogonal or "swept clean" of these eigenvector components. This is theoretically accomplished by setting

$$\{v_1^*\} = \{v_1\} - \sum_{j=1}^{f} [\{\tilde{X}_j\}^T[D]\{v_1\}]\{\tilde{X}_j\}, \tag{62}$$

where $\{v_1^*\}$ is the swept initial vector to be used in place of $\{v_1\}$ following its normalization. It can then be shown that all succeeding v-vectors generated by the recurrence algorithm (Equations 38 and 39) form a theoretically orthogonal set which does not contain components of $\{\tilde{X}_j\}$, j = 1, f. However, it has been shown (Reference 14) that the v-vectors degrade rapidly as the computations proceed, such that the later vectors are far removed from orthogonality to the earlier ones. This

---

**Previously calculated eigenvectors may be available from the following sources:
(1) The specification of fictitious free body supports on a SUPØRT card, which causes an equal number of rigid body modes to be automatically generated prior to entering an eigenvalue extraction routine.
(2) Checkpointing of previously obtained eigensolutions followed by a restart to obtain additional eigensolutions.

729

is caused by unavoidable computational round-off which, because of repeated multiplications by the unreduced eigenmatrix, $[\bar{B}]$, tends to amplify the contributions of the eigenvector components nearest the shift point in the calculated trial vectors. Thus, unless sufficiently accurate orthogonality of the trial vectors is maintained, they will be excessively rich in the modes near the shift point, and the solution of the reduced tridiagonal eigenvalue problem will yield a false bunching of eigenvalues around this point in the eigenspectrum.

To correct this problem, Gregory (Reference 6) experimented with the use of higher-precision computer operations, but found only marginal improvement in the final results. Later, Lanczos suggested a re-orthogonalization of the type

$$\{v^*_{i+1}\} = \{v_{i+1}\} - \sum_{j=1}^{i} [\{v_j\}^T[D]\{v_{i+1}\}]\{v_j\} - \sum_{j=1}^{f} [\{\tilde{X}_j\}^T[D]\{v_{i+1}\}]\{\tilde{X}_j\}, \qquad (63)$$

where $\{v_{i+1}\}$ is calculated by the unmodified recurrence algorithm and $\{v^*_{i+1}\}$ is an improved vector. While this improves matters substantially, it still does not eliminate the precision problem adequately. However, Ojalvo and Newman (Reference 8) found that the introduction of an iterative re-orthogonalization loop can make the trial vectors as orthogonal as necessary for extremely large systems. The procedure is as follows:

The vector $\{v_{i+1}\}$, obtained from either the recurrence algorithm or the pseudo-random number generator (see Section 10.6.2.4) is denoted as $\{v^{(0)}_{i+1}\}$ and re-orthogonalized with respect to all the previously obtained vectors. This is accomplished by iterating,

$$\begin{aligned}
\{v^{(1)}_{i+1}\} &= \{v^{(0)}_{i+1}\} - \sum_{j=1}^{i} [\{v_j\}^T[D]\{v^{(0)}_{i+1}\}]\{v_j\} - \sum_{j=1}^{f} [\{\tilde{X}_j\}^T[D]\{v^{(0)}_{i+1}\}]\{\tilde{X}_j\}, \\
\{v^{(2)}_{i+1}\} &= \{v^{(1)}_{i+1}\} - \sum_{j=1}^{i} [\{v_j\}^T[D]\{v^{(1)}_{i+1}\}]\{v_j\} - \sum_{j=1}^{f} [\{\tilde{X}_j\}^T[D]\{v^{(1)}_{i+1}\}]\{\tilde{X}_j\}, \\
&\vdots
\end{aligned} \qquad (64)$$

until an acceptable vector

$$\{v^{(s+1)}_{i+1}\} = \{v^{(s)}_{i+1}\} - \sum_{j=1}^{i} [\{v_j\}^T[D]\{v^{(s)}_{i+1}\}]\{v_j\} - \sum_{j=1}^{f} [\{\tilde{X}_j\}^T[D]\{v^{(s)}_{i+1}\}]\{\tilde{X}_j\}, \qquad (65)$$

is found which satisfies the orthogonality criteria

$$\max_{1 \le j \le i} \left| \{v_j\}^T [D] \{v_{i+1}^{(s)}\} \right| \le 10^{2-t}, \tag{66}$$

$$\max_{1 \le j \le f} \left| \{\tilde{X}_j\}^T [D] \{v_{i+1}^{(s)}\} \right| \le 10^{2-t}, \tag{67}$$

where t is the number of decimal digits carried by the computer. If, for some vector, the above criteria are not satisfied after a set number of iterations, then the program assumes that a new trial vector cannot be generated, and a reduced eigenvalue problem of order m = i is solved, as discussed in Section 10.6.2.3.

If the above criteria are met, then the resulting vector is normalized and set equal to the new normalized trial vector, i.e.,

$$\{v_{i+1}\} = \frac{\{v_{i+1}^{(s+1)}\}}{[\{v_{i+1}^{(s+1)}\}^T [D] \{v_{i+1}^{(s+1)}\}]^{1/2}} . \tag{68}$$

This new vector is used to compute the next off-diagonal term in the reduced tridiagonal matrix from the formula

$$d_{i+1} = \{v_{i+1}\}^T [B] \{v_i\}. \tag{69}$$

However, if

$$|d_{i+1}| \le 10^{2-t} \; |a_{i,i}| , \tag{70}$$

it is probable that $\{v_{i+1}\}$ is a null vector, possibly because the maximum number of linearly independent vectors, corresponding to the rank of the problem, has been exceeded. In this event, a reduced eigenvalue problem of order m = i is solved, as above.

If the criteria given by Equations 66, 67 and 70 are all met, then the new normalized vector $\{v_{i+1}\}$ is used to continue the reduction algorithm.

10.6.2.6 Criterion for the Negative Shift Parameter, α^2

Before tridiagonal reduction, the eigenvalue problem for natural frequencies in the neighborhood of zero is

$$[B]\{X\} = \Lambda\{X\}, \tag{71}$$

where

$$[B] = [C]^{-1}[M_{aa}]([C]^{-1})^T, \tag{72}$$

$$\{X\} = [C]^T\{\phi\} , \tag{73}$$

$$[K_{aa} + \alpha^2 M_{aa}] = [C][C]^T , \tag{74}$$

and

$$\Lambda = \frac{1}{\omega^2 + \alpha^2} . \tag{75}$$

The criterion involved in the choice of the shift parameter, α^2, is that it be large enough to render a possibly singular stiffness matrix non-singular (to the extent that Cholesky symmetric decomposition can be performed accurately in a finite digit computer), and small enough to prevent troublesome clustering of the eigenvalues, Λ_i. As an approach toward solving this problem, it is helpful to note that, given a symmetric positive-definite matrix [J], the result

$$[C][C]^T = [J] + [\delta J], \tag{76}$$

is obtained when it is factored, because of computer rounding errors. According to Wilkinson (Reference 15), the following inequality is almost always satisfied,

$$||\delta J||_1 \leq n(10^{1-t})\ ||J||_1, \tag{77}$$

where the above Holder-one matrix norms are equal to the maximum row sums of absolute values, n is the order of the matrix, and t is the number of decimal digits carried by the computer.

Thus, in order to render the matrix $[K_{aa} + \alpha^2 M_{aa}]$ non-singular, even when $[K_{aa}]$ itself is singular, some matrix norm of the modification $\alpha^2\ [M_{aa}]$ should be appreciably larger than the corresponding norm of $[\delta K_{aa}]$, where the latter matrix is in the "noise level" of the computer. On the basis of Equation 77, the minimum α^2 should therefore satisfy the requirement

$$\alpha^2_{min} \, ||M_{aa}||_1 \geq n(10^{1-t}) \, ||K_{aa}||_1 \, , \tag{78}$$

or

$$\alpha^2_{min} \geq n(10^{1-t}) \frac{\left(\max\limits_{1 \leq i \leq n} \sum\limits_{j=1}^{n} |k_{ij}| \right)}{\left(\max\limits_{1 \leq i \leq n} \sum\limits_{j=1}^{n} |m_{ij}| \right)} \tag{79}$$

where K_{ij} and m_{ij} are the elements of $[K_{aa}]$ and $[M_{aa}]$.

To further enhance the removal of possible singularities in the stiffness matrix, a factor of ten is applied to the right side and the maximum ratio of diagonal matrix elements is used. The resulting value of α^2_{min} is

$$\alpha^2_{min} = n(10^{2-t}) \left| \frac{K_{ii}}{m_{ii}} \right|_{max} ; \quad m_{ii} \neq 0. \tag{80}$$

A final requirement imposed on the shift parameter is that it be large enough to effect alterations in the last two-thirds of the significant digits in each diagonal term of the unmodified stiffness matrix, or equivalently,

$$\alpha^2_o = 10^{-t/3} \left| \frac{K_{ii}}{m_{ii}} \right|_{min} ; \quad m_{ii} \neq 0. \tag{81}$$

Consequently, the actual value of the shift used by the program is

$$\alpha^2 = \max(\alpha^2_{min}, \alpha^2_o). \tag{82}$$

If the resulting modified stiffness matrix is still singular, as indicated by a failure of the Cholesky decomposition process, the above value of α^2 is multiplied by a factor of 100 a maximum of two times in an attempt to render the undecomposed matrix non-singular. If this procedure fails, the problem execution is aborted due to unremovable stiffness matrix singularities.

10.6.2.7 Error Bounds on the Computed Eigenvalues

Once the modes of the reduced problem have been extracted, close upper and lower bounds on the eigenvalue errors can be obtained rather economically. This provides the user with an a posteriori check on the number of accurately calculated eigenvalues which, in most cases, will be greater than the number requested.

In carrying out the development, it is convenient to write the eigenproblem in its inverse form

$$[\bar{B}]\{X\} = \Lambda\{X\}, \tag{83}$$

where

$$\Lambda = \frac{1}{\lambda_a - \lambda_o}. \tag{84}$$

Considering some approximation, $\{\bar{X}_i\}$, to an exact eigenvector, $\{X_i\}$, it can be expanded in terms of the exact modal matrix as follows:

$$\{\bar{X}_i\} = [Z][b_i], \tag{85}$$

where

$$\underset{n\times n}{[Z]} = [\{X_1\} \vdots \{X_2\} \vdots \ldots . \vdots \{X_n\}], \tag{86}$$

and

$$\{b_i\} = \begin{Bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{Bmatrix} \tag{87}$$

is a vector of scalar coefficients.

If the eigenvalue approximation associated with $\{\bar{X}_i\}$ is denoted by $\bar{\Lambda}_i$, then the residual vector for the i^{th} modal approximation is given by

$$\begin{aligned} \{R_i\} &= [\bar{B} - \bar{\Lambda}_i[I]]\{\bar{X}_i\} \\ &= [\bar{B} - \bar{\Lambda}_i[I]][Z]\{b_i\}. \end{aligned} \tag{88}$$

However

$$[\bar{B}][Z] = [Z][\ \Lambda_d\ ], \tag{89}$$

where $[\ \Lambda_d\ ]$ is the diagonal matrix of eigenvalues, Λ_i, i = 1, n.

It follows that

$$\{R_i\} = [Z][[\,\Lambda_d\,] - \bar{\Lambda}_i[I]]\{b_i\}. \tag{90}$$

Denoting a weighted root-mean-square residual by

$$r_i = (\{R_i\}^T[D]\{R_i\})^{1/2}, \tag{91}$$

it can be seen that

$$r_i^2 = \{b_i\}^T[[\,\Lambda_d\,] - \bar{\Lambda}_i[I]][Z]^T[D][Z][[\,\Lambda_d\,] - \bar{\Lambda}_i[I]]\{b_i\}. \tag{92}$$

However, without any loss of generality, the eigenvectors can be orthonormalized so that

$$\{X_j\}^T[D]\{X_i\} = \left\{\begin{matrix} 1; & i = j \\ 0; & i \neq j \end{matrix}\right\}. \tag{93}$$

The modal matrix is therefore unitary, i.e.,

$$[Z]^T[D][Z] = [I]\,, \tag{94}$$

and

$$\begin{aligned} r_i^2 &= \{b_i\}^T[[\,\Lambda_d\,] - \bar{\Lambda}_i[I]]^2\{b_i\} \\ &= \sum_{j=1}^{n} b_{ji}^2\,(\Lambda_j - \bar{\Lambda}_i)^2, \end{aligned} \tag{95}$$

where $\{b_{ji}\}$ refers to the j^{th} element of the vector $\{b_i\}$.

At this point it should be noted that the program also normalizes the eigenvectors $\{y_i\}$ of the reduced tridiagonal problem, so that

$$\{\bar{X}_i\}^T[D]\{\bar{X}_i\} = \{y_i\}^T[V]^T[D][V]\{y_i\} = 1, \tag{96}$$

where

$$[V] = [\{v_1\} \,\vdots\, \{v_2\} \,\vdots\, \ldots. \,\vdots\, \{v_m\}]. \tag{97}$$

Thus

$$\{\bar{X}_i\}^T[D]\{\bar{X}_i\} = \sum_{j=1}^{n} b_{ji}^2 = 1, \tag{98}$$

and it follows that

$$r_i^2 \geq \min_{1 \leq j \leq n} (\Lambda_j - \bar{\Lambda}_i)^2. \tag{99}$$

Therefore, when $\bar{\Lambda}_i$ is close to an exact eigenvalue, Λ_i, it can be seen that

$$|\Lambda_i - \bar{\Lambda}_i| \le |r_i|, \tag{100}$$

or

$$|(\frac{\bar{\lambda}_{ai} - \lambda_o}{\lambda_{ai} - \lambda_o}) - 1| \le |\frac{r_i}{\bar{\Lambda}_i}|. \tag{101}$$

Since $\bar{\lambda}_{ai}$ is close to λ_{ai} when $\bar{\Lambda}_i$ is close to Λ_i, Equation 101 implies that

$$|1 - \frac{\bar{\lambda}_{ai}}{\lambda_{ai}}| \le |\frac{r_i}{\bar{\Lambda}_i(1 + \lambda_o \bar{\Lambda}_i)}| , \tag{102}$$

which is a measure of the maximum relative error in the i^{th} physical eigenvalue*.

The residual, r_i, in the above equation can be evaluated quite easily via the following approach.

Set

$$\{\bar{X}_i\} = [V]\{y_i\} \tag{103}$$

in Equation 88. Then

$$\{R_i\} = [[\bar{B}][V] - \bar{\Lambda}_i[V]]\{y_i\}. \tag{104}$$

However, from Equation 36, it is seen that

$$[\bar{B}][V] = [V][A] + d_{m+1}[G_{m+1}], \tag{105}$$

where [A] is the tridiagonal reduced matrix and

$$[G_{m+1}] = [\{0\} \mid \{0\} \mid \{0\} \mid \{v_{m+1}\}]. \tag{106}$$

Therefore

$$\{R_i\} = [[V][A] + d_{m+1}[G_{m+1}] - \bar{\Lambda}_i[V]]\{y_i\} . \tag{107}$$

---

*This test is obviously invalid when $\lambda_{ai} = 0$, i.e., a rigid body solution. In this event, the computed value of $\bar{\lambda}_{ai}$ is, in itself, a measure of the absolute error in the physical eigenvalue and no further accuracy information is needed. The existence of an eigenvalue as an additional rigid body mode not requested on a SUPØRT card in the Bulk Data deck is detected by the criterion $|\bar{\lambda}_{ai}| < 10^{-t/3}$, in which case Equation 102 is bypassed.

But

$$[A]\{y_i\} = \bar{\Lambda}_i\{y_i\}, \tag{108}$$

so that

$$\{R_i\} = d_{m+1}[G_{m+1}]\{y_i\}, \tag{109}$$

and

$$r_i^2 = \{R_i\}^T[D]\{R_i\} = d_{m+1}^2\{y_i\}^T[G_{m+1}]^T[D][G_{m+1}]\{y_i\}. \tag{110}$$

It can be seen that

$$[G_{m+1}]^T[D][G_{m+1}] = \begin{bmatrix} 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & 1 \end{bmatrix}, \tag{111}$$

so that finally

$$r_i^2 = d_{m+1}^2 y_{mi}^2, \tag{112}$$

where y_{mi} is the last element of the vector $\{y_i\}$.

Therefore, Equation 102 assumes the form

$$\left|1 - \frac{\bar{\lambda}_{ai}}{\lambda_{ai}}\right| \le \frac{|d_{m+1} \cdot y_{mi}|}{|\bar{\Lambda}_i(1 + \lambda_0\bar{\Lambda}_i)|} \tag{113}$$

Thus, it is seen that the eigenvalue errors are proportional to d_{m+1}, which is the next off-diagonal term that would be generated, had the reduced tridiagonal matrix [A] been increased from order m to order m+1. Equation 113 shows that this term is further modified by a weighting factor y_{mi}, which is the last term in the reduced-system eigenvector associated with $\bar{\lambda}_{ai}$.

The use of the above error bound formula as a criterion for selecting acceptable eigensolutions is described in Section 10.6.3.

10.6.3 Summary of Computational Procedures and Flow Charts - Real Eigenvalue Analysis

Flow diagrams illustrating the computational procedures are shown in Figures 1 and 2. The details of each block are summarized below.

1. Calculate Small Negative Shift Parameters, α^2, (see Section 10.6.2.6)

In the case of unshifted vibration mode problems, the negative shift parameter for removing possible singularities is found from:

$$\alpha^2 = \max(\alpha^2_{min}, \alpha^2_o), \tag{114}$$

where

$$\alpha^2_{min} = n(10^{2-t}) \left|\frac{K_{ii}}{m_{ii}}\right|_{max}; \quad m_{ii} \neq 0, \tag{115}$$

and

$$\alpha^2_o = 10^{-t/3}\left|\frac{K_{ii}}{m_{ii}}\right|_{min}; \qquad m_{ii} \neq 0. \tag{116}$$

K_{ii} and m_{ii} are the diagonal elements of $[K_{aa}]$ and $[M_{aa}]$, respectively, n is the number of $\{u_a\}$ degrees of freedom, and t is the number of decimal digits carried by the computer.

2. Zero-Out Excessively Small Elements of [M] Matrix (see Section 10.6.2.3)

   a. Compare the magnitudes of all off-diagonal elements of [M] with the corresponding diagonal elements to determine whether

$$\left|\frac{m_{ij}}{m_{ii}}\right| \leq 10^{-2t/3}; \quad i \neq j,\ m_{ii} \neq 0. \tag{117}$$

   b. Set $m_{ij} = 0.0$ for every off-diagonal element satisfying the above criterion.

3. Establish Tentative Reduced Problem Size (see Section 10.6.2.3)

   a. Count the number, $\bar{n}$, of non-null columns or rows in the above modified [M] matrix and set

$$\bar{r} = \bar{n} - f, \tag{118}$$

   where f is the number of previously computed eigensolutions.

b. Calculate a tentative size, m, of the reduced eigenproblem from

$$m = \min[2\bar{q} + 10, \bar{r}], \tag{119}$$

where

$$\bar{q} = q - f, \tag{120}$$

and q is the total number of accurate eigenvalues requested by the user, including previously computed modes. If $\bar{q} > \bar{r}$, the program will try to find all existing solutions.

4. Construct Factors of $[\bar{K}]$ Matrix (see Section 10.6.2.1)

a. Set

(i) $$[\bar{K}] = [K_{aa} - \omega_o^2 M_{aa}], \quad \text{(Shifted Vibration Mode Problems)} \tag{121}$$

or

(ii) $$[\bar{K}] = [K_{aa} + \alpha^2 M_{aa}], \quad \text{(Unshifted Vibration Mode Problems)} \tag{122}$$

or

(iii) $$[\bar{K}] = [K_{aa}]. \quad \text{(Buckling Problems)} \tag{123}$$

b. Perform a non-square root decomposition:

$$[\bar{K}] = [L][\lceil d \rfloor][L]^T, \tag{124}$$

for case (i), or a Cholesky symmetric decomposition:

$$[\bar{K}] = [C][C]^T \tag{125}$$

for cases (ii) and (iii), using real arithmetic without pivoting. Save the triangular and diagonal factors. If the decomposition for case (i) fails or the decompositions for cases (ii) and (iii) fail after two increases in α^2 by factors of one hundred, then the problem execution is aborted because of unremovable stiffness matrix singularities.

5. Execute Tridiagonal Reduction Algorithm (see flow diagram for this block, Figure 2)

5.1 Initialize the Recurrence Algorithm (see Section 10.6.2.2)

Initialize the vector index to i = 0 and set

$$\{v_o\} = \{0\}, \tag{126}$$

where $\{v_o\}$ is an $(n \times 1)$ null vector.

5.2 Generate a Starting or Restart Vector and Set $d_{i+1} = 0.0$ (see Section 10.6.2.4)

a. Construct an n-element vector $\{w\}$ using a pseudo-random number generator.

b. Solve for an unnormalized trial vector from the equation

$$\{\bar{v}_{i+1}\} = [\bar{B}]\{w\}, \tag{127}$$

where

$$[\bar{B}] = ([L]^{-1})^T[\lceil d \rfloor]^{-1}[L]^{-1}[M_{aa}], \text{ (case i)} \tag{128}$$

or

$$[\bar{B}] = [C]^{-1}[M_{aa}]([C]^{-1})^T \;, \qquad \text{(case ii)} \tag{129}$$

or

$$[\bar{B}] = [C]^{-1}[K^d_{aa}]([C]^{-1})^T . \qquad \text{(case iii)} \tag{130}$$

Forward and backward passes are used to perform the above inverse operations.

c. Normalize the above vector:

$$\{v^{(o)}_{i+1}\} = \left[\frac{1}{\{\bar{v}_{i+1}\}^T[D]\{\bar{v}_{i+1}\}}\right]^{1/2}\{\bar{v}_{i+1}\}, \tag{131}$$

where

$$[D] = [M_{aa}], \quad \text{(case i)} \tag{132}$$

$$[D] = [I] \;. \quad \text{(cases ii and iii)} \tag{133}$$

d. Set $d_{i+1} = 0.0$ and proceed to block 5.5.

5.3 Create One Approximate Trial Vector and One Diagonal Coefficient (see Section 10.6.2.2)

The recurrence algorithm is:

$$a_{i,i} = \{v_i\}^T[B]\{v_i\}, \tag{134}$$

$$\{\bar{v}_{i+1}\} = [\bar{B}]\{v_i\} - a_{i,i}\{v_i\} - d_i\{v_{i-1}\}, \tag{135}$$

$$\bar{d}_{i+1} = [\{\bar{v}_{i+1}\}^T[D]\{\bar{v}_{i+1}\}]^{1/2}, \tag{136}$$

$$\{v_{i+1}^{(0)}\} = \frac{1}{\bar{d}_{i+1}} \{\bar{v}_{i+1}\}, \tag{137}$$

where

$$[B] = [D][\bar{B}], \tag{138}$$

and $\{\bar{v}_{i+1}^{(0)}\}$ is an approximation to the new trial vector.

5.4 First Normalization Test (see Section 10.6.2.4)

The test is

$$|\bar{d}_{i+1}| \geq 10^{2-t} |a_{i,i}|. \tag{139}$$

Pass: Proceed direction to block 5.5.

Fail: Return to block 5.2, generate a new restart vector for $\{v_{i+1}^{(0)}\}$, and proceed to block 5.5.

5.5 Iterate to Obtain Orthogonalized Vector (see Section 10.6.2.5)

Designate $\{\tilde{X}_j\}$, $j = 1, f$ as previously calculated and stored eigenvectors. Perform the iterations,

$$\{v_{i+1}^{(s+1)}\} = \{v_{i+1}^{(s)}\} - \sum_{j=1}^{i} [\{v_j\}^T[D]\{v_{i+1}^{(s)}\}]\{v_j\}$$

$$- \sum_{j=1}^{f} [\{\tilde{X}_j\}^T[D]\{v_{i+1}^{(s)}\}]\{\tilde{X}_j\}; \quad s = 0, 1, 2\ldots, \tag{140}$$

until

$$\max_{1 \le j \le i} |\{v_j\}^T[D]\{v_{i+1}^{(s)}\}| \le 10^{2-t},$$

and (141)

$$\max_{1 \le j \le f} |\{\tilde{X}_j\}^T[D]\{v_{i+1}^{(s)}\}| \le 10^{2-t},$$

or

$$s = 14.$$

If the orthogonality criterion, Equation 141, is satisfied, proceed to block 5.6. Otherwise, set the problem size, m, equal to i and proceed to Exit.

5.6 Normalize the Orthogonalized Trial Vector (see Section 10.6.2.5)

Compute

$$\{v_{i+1}\} = \frac{\{v_{i+1}^{(s+1)}\}}{[\{v_{i+1}^{(s+1)}\}[D]\{v_{i+1}^{(s+1)}\}]^{1/2}} \tag{142}$$

This is the new orthogonalized and normalized trial vector.

5.7 Second Normalization Test and Creation of Off-Diagonal Coefficient (see Section 10.6.2.5)

a. Compute the next off-diagonal term in the reduced tridiagonal matrix from

$$d_{i+1} = \{v_{i+1}\}^T[B]\{v_i\}. \tag{143}$$

b. Verify whether the following test is met:

$$|d_{i+1}| \le 10^{2-t} |a_{i,i}|. \tag{144}$$

If it is, set i = i+1 and return to block 5.3 for continuation of the recurrence algorithm. If the test fails, set m = i to reduce the problem size and proceed to Exit. Only i modes can be obtained, since more than r - f modes may have been requested.

6. Solve Reduced-System Eigenvalue Problem (see Section 10.6.2.2)

a. The coefficients a_{11}, a_{22}, ..., a_{mm} and d_2, d_3, ..., d_m, computed in block 5 (of

Figure 1) are interpreted as the following symmetric, tridiagonal array:

$$[A] = \begin{bmatrix} a_{11} & d_2 & & & \\ d_2 & a_{22} & d_3 & & \\ & d_3 & a_{33} & \ddots & \\ & & \ddots & \ddots & d_m \\ & & & d_m & a_{mm} \end{bmatrix} . \qquad (145)$$

b. The m^th^ order eigenvalue problem

$$[A]\{y\} = \bar{\Lambda}\{y\}, \qquad (146)$$

is solved for the eigenvalues, $\bar{\Lambda}_i$, and eigenvectors $\{y_i\}$ using a Q-R algorithm and eigenvector computational procedure similar to that described in Sections 10.2.3 and 10.2.4.

c. The reduced system eigenvectors are normalized so that

$$\{y_i\}^T\{y_i\} = 1; \quad i = 1, m. \qquad (147)$$

7. Compute Maximum Eigenvalue Errors (see Section 10.6.2.7)

a. The maximum absolute relative errors in the computed physical eigenvalues are obtained from

$$\xi_i = \frac{|(\bar{d}_{m+1})(y_{mi})|}{|\bar{\Lambda}_i(1 + \lambda_o\bar{\Lambda}_i)|}; \quad i = 1, m, \qquad (148)$$

where $\bar{d}_{m+1}$ is the last off-diagonal term computed in block 5.3 and y_{mi} is the last element in the vector $\{y_i\}$. If the physical eigenvalue, $\frac{1}{\bar{\Lambda}_i} + \lambda_o$, corresponds to a rigid body mode, the above computation is invalid and therefore bypassed. A rigid body mode is assumed to occur whenever

$$\left|\frac{1}{\bar{\Lambda}_i} + \lambda_o\right| \leq 10^{-t/3} \qquad (149)$$

and is denoted by setting the relative error, ξ_i, equal to an exact zero.

b. The eigenvalues are processed in order of increasing distance from the center of range of interest, λ_o, to determine whether their associated ξ_i values meet an acceptable relative error tolerance set by the user on the EIGR or EIGB bulk data card (the default value is 0.001/n where n is the order of the unreduced problem). The first eigenvalue not meeting the tolerance test, as well as all subsequent eigenvalues further removed from the center of interest, are considered to lack sufficient accuracy and are therefore rejected.

c. Acceptable eigenvalues obtained in the above manner are reordered in terms of increasing physical value for subsequent processing by the program.

8. Compute Physical Eigenvalues and Eigenvectors (see Section 10.6.2.2)

The mathematical eigenvalues, $\bar{\Lambda}_i$, and eigenvectors, $\{y_i\}$, are converted to physical form as follows:

$$\bar{\lambda}_i = -\frac{1}{\bar{\Lambda}_i}, \qquad \text{(buckling problems)} \tag{150}$$

$$\bar{\omega}_i^2 = \frac{1}{\bar{\Lambda}_i} - \alpha^2, \qquad \text{(unshifted vibration mode problems)} \tag{151}$$

$$\bar{\omega}_i^2 = \frac{1}{\bar{\Lambda}_i} + \omega_o^2, \qquad \text{(shifted vibration mode problems)} \tag{152}$$

$$\{\bar{\phi}_i\} = ([C]^{-1})^T[V]\{y_i\}, \qquad \text{(buckling or unshifted vibration mode problems)} \tag{153}$$

$$\{\bar{\phi}_i\} = [V]\{y_i\}, \qquad \text{(shifted vibration mode problems)} \tag{154}$$

where

$$[V] = [\{v_1\}, \{v_2\}, \ldots, \{v_m\}]. \tag{155}$$

10.6.4 Theory for Complex Eigenvalue Analysis

10.6.4.1 Problem Formulation

The general complex eigenvalue problem is stated in the form

$$[Mp^2 + Bp + K]\{u\} = 0 \quad , \tag{156}$$

where the matrices [M], [B] and [K] may be real or complex, symmetric or unsymmetric, and singular or non-singular. A specified number of eigenvalues, p, lying closest to a specified point, λ_o, (called a shift point) in the complex plane are to be found, as well as the associated eigenvectors {u}. The eigenvalues may include multiplicities.

Following a development similar to that of Section 10.4.4.1, a velocity vector

$$\{v\} = p\{u\} \quad , \tag{157}$$

and a shifted eigenvalue

$$\lambda = p - \lambda_o \quad , \tag{158}$$

are first defined.

Substitution of Equations 157 and 158 into Equation 156 gives

$$\left[\begin{array}{c:c} K & B+\lambda_o M \\ \hdashline -\lambda_o I & I \end{array}\right]\left\{\begin{array}{c} u \\ \hdashline v \end{array}\right\} = \lambda \left[\begin{array}{c:c} 0 & -M \\ \hdashline I & 0 \end{array}\right]\left\{\begin{array}{c} u \\ \hdashline v \end{array}\right\} \quad , \tag{159}$$

or, using the inverse form,

$$[A]\{x\} = \Lambda\{x\}, \tag{160}$$

where

$$[A] = \left[\begin{array}{c:c} K & B+\lambda_o M \\ \hdashline -\lambda_o I & I \end{array}\right]^{-1} \left[\begin{array}{c:c} 0 & -M \\ \hdashline I & 0 \end{array}\right], \tag{161}$$

$$\{x\} = \left\{\begin{array}{c} u \\ \hdashline v \end{array}\right\} \quad , \tag{162}$$

and

$$\Lambda = \frac{1}{p - \lambda_o} . \tag{163}$$

Equation 160 is an eigenvalue problem in standard form, showing that the order of the eigenvalue problem is doubled due to the presence of the [B] matrix. In the special case where [B] is null (i.e. no damping), the problem formulation becomes

$$[Mp^2 + K]\{u\} = 0, \tag{164}$$

and the double-size eigenvalue problem can be avoided by considering the mathematical eigenvalue to be p^2. Thus, let

$$\lambda^2 = p^2 - \lambda_o^2 , \tag{165}$$

and

$$\Lambda = \frac{1}{\lambda^2} . \tag{166}$$

Substitution of Equations 165 and 166 into Equation 164 and the use of the inverse form result in

$$[K+\lambda_o{}^2 M]^{-1}[-M]\{u\} = \Lambda\{u\}. \tag{167}$$

Comparing the above with Equation 160 shows that the standard form with a null [B] matrix implies that

$$[A] = [K+\lambda_o{}^2 M]^{-1}[-M], \tag{168}$$

and

$$\{x\} = \{u\}. \tag{169}$$

Since the eigenmatrix [A] is, in general, unsymmetric, the eigenvectors {x} are orthogonal to the eigenvectors, $\{\bar{\bar{x}}\}$, of the transpose eigenproblem

$$[A]^T\{\bar{\bar{x}}\} = \Lambda\{\bar{\bar{x}}\}, \tag{170}$$

so that for $\Lambda_i \neq \Lambda_j$,

$$\{\bar{\bar{x}}_j\}^T\{x_i\} = 0; \quad i \neq j. \tag{171}$$

The above relationship is a biorthogonality condition and the associated eigenvectors $\{x_i\}$ and $\{\bar{\bar{x}}_j\}$ are called right and left eigenvectors, respectively.

10.6.4.2 The Reduction Algorithm

A reduction of the order of the eigenvalue problem, Equation 160, is effected through the transformation

$$\underset{n\times 1}{\{\hat{x}\}} = \underset{n\times m}{[V]}\ \underset{m\times 1}{\{y\}}, \tag{172}$$

and

$$\underset{n\times 1}{\{\hat{\hat{x}}\}} = \underset{n\times m}{[\bar{\bar{V}}]}\ \underset{m\times 1}{\{\bar{\bar{y}}\}}, \tag{173}$$

where $\{\hat{x}\}$ and $\{\hat{\hat{x}}\}$ are approximations of $\{x\}$ and $\{\bar{\bar{x}}\}$, respectively, n is the order of the unreduced problem, and $m \leq n$. The above transformation matrices are chosen to be biorthonormal, so that

$$[\bar{\bar{V}}]^T[V] = [I]. \tag{174}$$

From Equations 160, 172, 173, and 174, it is seen that

$$[H]\{y\} = \bar{\Lambda}\{y\}, \tag{175}$$

where

$$\underset{m\times m}{[H]} = [\bar{\bar{V}}]^T[A][V], \tag{176}$$

and $\bar{\Lambda}$ is an approximation of the eigenvalue, Λ.

Thus, Equation 175 is an mth order eigenvalue problem, where $m \le n$. The value of m is established according to the criteria given in Section 10.6.4.3.

As in the case of real eigenvalue analysis (see Section 10.6.2.2) the Lanczos algorithm is used to construct the transformation matrices vector by vector, i.e.,

$$\underset{n\mathrm{x}m}{[V]} = [\{v_1\}, \{v_2\}, \ldots\ldots, \{v_m\}], \tag{177}$$

$$\underset{n\mathrm{x}m}{[\bar{\bar{V}}]} = [\{\bar{\bar{v}}_1\}, \{\bar{\bar{v}}_2\}, \ldots\ldots, \{\bar{\bar{v}}_m\}], \tag{178}$$

such that the reduced mxm matrix [H] is tridiagonal and its eigenvalues accurately approximate the roots of Equation 160 having the largest magnitude (or, equivalently, the physical roots, p, closest to the specified point of interest, λ_o, in the complex plane).

The form of the algorithm for generating successive vector pairs is, according to the Lanczos technique,

$$d_{i+1}\{v_{i+1}\} = \{w_{i+1}\} = [A]\{v_i\} - a_{i,1}\{v_1\} - a_{i,2}\{v_2\} \ldots\ldots - a_{i,i}\{v_i\}, \tag{179}$$

$$\bar{\bar{d}}_{i+1}\{\bar{\bar{v}}_{i+1}\} = \{\bar{\bar{w}}_{i+1}\} = [A]^T\{\bar{\bar{v}}_i\} - \bar{\bar{a}}_{i,1}\{\bar{\bar{v}}_1\} - \bar{\bar{a}}_{i,2}\{\bar{\bar{v}}_2\} \ldots\ldots - \bar{\bar{a}}_{i,i}\{\bar{\bar{v}}_i\}, \tag{180}$$

where $\{v_1\}$ and $\{\bar{\bar{v}}_1\}$ are starting vectors (see Section 10.6.4.4), $1 \le i \le m-1$, and all m vector pairs are biorthonormal. This implies that

$$a_{i,j} = \{\bar{\bar{v}}_j\}^T[A]\{v_i\}, \tag{181}$$

and

$$\bar{\bar{a}}_{i,j} = \{v_j\}^T[A]^T\{\bar{\bar{v}}_i\}, \tag{182}$$

while the normalizing factors are given by

$$d_{i+1} = \bar{\bar{d}}_{i+1} = [\{\bar{\bar{w}}_{i+1}\}^T\{w_{i+1}\}]^{1/2}. \tag{183}$$

Through the use of symmetry arguments similar to those employed in Section 10.6.2.2 for real eigenvalue analysis, it can be shown that

$$a_{i,i} = \bar{\bar{a}}_{i,i}, \tag{184}$$

$$a_{i,i-1} = d_i = \bar{\bar{a}}_{i,i-1}, \tag{185}$$

and

$$a_{i,j} = \bar{\bar{a}}_{i,j} = 0; \quad j<i-1. \tag{186}$$

The transformed, reduced eigenmatrix in Equation 175 is consequently tridiagonal and symmetric, having the form

$$[H] = [\bar{V}]^T[A][V] = \begin{bmatrix} a_{11} & d_2 & & & \\ d_2 & a_{22} & d_3 & & \\ & d_3 & a_{33} & d_4 & \\ & & \ddots & \ddots & d_m \\ & & & d_m & a_{mm} \end{bmatrix} \tag{187}$$

The matrix coefficients are theoretically given by the simplified recurrence formulas

$$\left.\begin{aligned} a_{i,i} &= \{\bar{\bar{v}}_i\}^T[A]\{v_i\} \\ \{w_{i+1}\} &= [A]\{v_i\} - a_{i,i}\{v_i\} - d_i\{v_{i-1}\} \\ \{\bar{\bar{w}}_{i+1}\} &= [A]^T\{\bar{\bar{v}}_i\} - a_{i,i}\{\bar{\bar{v}}_i\} - d_i\{\bar{\bar{v}}_{i-1}\} \\ d_{i+1} &= [\{\bar{\bar{w}}_{i+1}\}^T\{w_{i+1}\}]^{1/2} \end{aligned}\right\} \quad i = 1, m \tag{188}$$

$$\left.\begin{aligned} \{v_{i+1}\} &= \frac{1}{d_{i+1}}\{w_{i+1}\} \\[2ex] \{\bar{\bar{v}}_{i+1}\} &= \frac{1}{d_{i+1}}\{\bar{\bar{w}}_{i+1}\} \end{aligned}\right\} \quad i = 1, m-1 \tag{189}$$

where the sequence is initialized by choosing random, biorthonormal starting vectors for $\{v_1\}$, $\{\bar{\bar{v}}_1\}$ and setting $d_1 = 0$; $\{v_o\} = \{\bar{\bar{v}}_o\} = \{0\}$.

The final off-diagonal term, d_{m+1}, given by Equations 188 is used in establishing error estimates for the computed eigenvalues (see Section 10.6.4.6). In addition, the above algorithm is modified in the computational scheme as follows:

1. Each pair of vectors $\{v_{i+1}\}$, $\{\bar{\bar{v}}_{i+1}\}$, calculated in Equations 189 is reorthogonalized to all previously computed pairs, as described in Section 10.6.4.5, before re-entering Equations 188.

2. The size, m, of the reduced problem is a function of the number of accurate eigenvalues requested by the user and is limited to the number of finite physical eigenvalues available (see Section 10.6.4.3).

The eigenvalues, $\bar{\Lambda}$, and eigenvectors, $\{y\}$, of Equation 175 are extracted using the Q-R iteration algorithm and eigenvector computational scheme described in connection with the Upper Hessenberg method (Section 10.5). They are then converted to physical form as follows:

$$\left.\begin{aligned} \bar{p}_i &= \frac{1}{\bar{\Lambda}_i} + \lambda_o \\[2ex] \left\{\begin{array}{c} u \\ \hline v \end{array}\right\}_i &= [V]\{y_i\} \end{aligned}\right\} \quad [B] \neq [0], \tag{190}$$

$$\left.\begin{aligned} \bar{p}_i &= \left(\frac{1}{\bar{\Lambda}_i} + \lambda_o^{\,2}\right)^{1/2}; \quad \mathrm{Im}(\bar{p}_i) > 0 \\[2ex] \{u_i\} &= [V]\{y_i\} \end{aligned}\right\} \quad [B] = [0]. \tag{191}$$

The velocity vector $\{v_i\}$ in Equation 190 is discarded prior to further processing of the set of eigensolutions by the program. Also, all solutions which fail the FEER error test are rejected. However, the number of acceptable solutions will, in all probability, equal or exceed the number requested by the user when the reduced problem size is chosen according to the criteria described in the following section.

10.6.4.3 Criteria for the Size of the Reduced Eigenvalue Problem

The maximum number of finite eigensolutions, including any existing rigid body modes, is equal to the rank, r, of the eigenmatrix [A] in Equation 160. Thus, for example, massless degrees of freedom, appearing as zero diagonal terms in the [M] matrix, will result in singularities (rank reduction), which imply infinite physical eigenvalues. These spurious roots are swept out of the problem in the complex FEER process (see Section 10.6.4.4) with a consequent reduction in the available eigensolutions.

A further consideration in limiting the maximum problem size is that the user has the option of requesting eigensolutions in the neighborhood of several shift points $(\lambda_{01}, \lambda_{02}, \ldots)$ in the complex plane. In the Tridiagonal Reduction method, all eigensolutions, f, obtained for previous shift points are swept out of the problem to prevent their re-generation when dealing with the current shift point (see Section 10.6.4.5). This implies that the maximum possible size, m, of the reduced problem is further limited to

$$m_{max} = r - f. \tag{192}$$

On the basis of numerical experiments, similar to those cited in Section 10.6.2.3 for real eigenvalue analysis, it has been found that when $m << m_{max}$, a first grouping of more than $m/2$ computed eigenvalues closest to the shift point are in accurate agreement with the corresponding number of exact eigenvalues, provided that $7 \leq m \leq m_{max}$. The remaining reduced-system roots are spread across the remaining exact eigenspectrum. To enhance the accuracy of the associated eigenvectors, the minimum problem size is further increased to twelve, again assuming that $m << m_{max}$.

Thus, if the user requests a total of $\bar{q}$ eigenvalues closest to a specified point in the complex plane, the order of the reduced problem is initially set to

$$m = \min\,[(2\bar{q}+10),(2n-f)]; \quad [B] \neq [0], \tag{193}$$

$$= \min\,[(2\bar{q}+10),\ (n-f)]; \quad [B] = [0]. \tag{194}$$

Although the total number of eigensolutions requested should not exceed m_{max}, there is usually no simple way to discern this upper limit in complex eigenvalue problems. However, the reorthogonalization tests of Section 10.6.4.5 are designed to automatically establish this upper limit. If the latter tests fail for some vector pair $\{v_{i+1}\}$, $\{\bar{\bar{v}}_{i+1}\}$, this is an indication that a null vector has been generated because m_{max} linearly independent vectors have already been obtained. The recurrence algorithm, Equations 188 and 189, is then terminated and the order of the eigenproblem is further reduced to $m = i$.

10.6.4.4 Choice of the Initial Trial Vector and Restart Vectors

Because of the inverse relationship between the computed eigenvalues, $\bar{\Lambda}$, and the physical eigenvalues, $\bar{p}$, (see Equations 190 and 191), spurious eigenvalues corresponding to $\bar{\Lambda} = 0$ are equivalent to $\bar{p} \to \infty$. Since these eigenvalues and their corresponding eigenvectors are of no interest and may cause numerical instabilities, they are eliminated from the reduced tridiagonal problem by employing a constrained set of $(v, \bar{\bar{v}})$ vector pairs having zero projection on the set of eigenvectors associated with $\Lambda = 0$. Extending the argument developed in Section 10.6.2.4 for real eigenvalue analysis, any non-null, right and left vectors $\{w_1\}$, $\{\bar{\bar{w}}_1\}$, generated from any other non-null vectors $\{w_r\}$, $\{\bar{\bar{w}}_r\}$ through the transformations

$$\{w_1\} = [A]\{w_r\}, \tag{195}$$

$$\{\bar{\bar{w}}_1\} = [A]^T\{\bar{\bar{w}}_r\}, \tag{196}$$

will contain no components of the eigenvectors corresponding to $\bar{\Lambda} = 0$. Further, after the above vectors are orthonormalized

$$\{v_1\} = \frac{1}{[\{\bar{\bar{w}}_1\}^T\{w_1\}]^{1/2}}\{w_1\} \; . \tag{197}$$

$$\{\bar{\bar{v}}_1\} = \frac{1}{[\{\bar{\bar{w}}_1\}^T\{w_1\}]^{1/2}}\{\bar{\bar{w}}_1\}, \tag{198}$$

the next vectors generated,

$$\{w_2\} = [A]\{v_1\} - a_{11}\{v_1\}, \tag{199}$$

$$\{\bar{\bar{w}}_2\} = [A]^T\{\bar{\bar{v}}_1\} - a_{11}\{\bar{\bar{v}}_1\}, \tag{200}$$

as well as all subsequent trial vectors will be free of $\bar{\Lambda} = 0$ eigenvectors.

Thus, the starting vectors, $\{v_1\}$, $\{\bar{\bar{v}}_1\}$, in the reduction algorithm are generated from Equations 195 through 198, starting with pseudo-random vectors $\{w_r\}$ and $\{\bar{\bar{w}}_r\}$. If these pseudo-random vectors should, by chance, be deficient in some true eigenvector components, then a null vector $\{v_{i+1}\}$ or $\{\bar{\bar{v}}_{i+1}\}$ may be generated at some point in Equations 188 and 189. This is indicated by the appearance of an off-diagonal term, d_{i+1}, which is exceedingly small compared to the corresponding diagonal term, $a_{i,i}$, in the reduced tridiagonal matrix. The test used is that a null vector has been generated when

$$|d_{i+1}| \leq 10^{-t/2}|a_{i,i}|, \tag{201}$$

where t is the number of decimal digits being carried in the computations. In this event, new restart vectors are employed for $\{w_{i+1}\}$, $\{\bar{\bar{w}}_{i+1}\}$. These vectors are generated exactly as in the case of the initial trial vectors, but using different pseudo-random number seeds. The recurrence algorithm for the generation of subsequent vector pairs $\{v_{i+1}\}$, $\{\bar{\bar{v}}_{i+1}\}$, etc., is then continued in the usual manner until the required number of vector pairs have been generated.

10.6.4.5 Sweeping-out of Previously Obtained Eigenvectors and Reorthogonalization of the Trial Vectors

As in the case of real eigenvalue analysis, successive trial vectors generated by the theoretical algorithm, Equations 188 and 189, tend to degrade rapidly as the computations proceed in a finite digit computer, such that later right vectors are far removed from orthogonality to earlier left vectors, and previously computed eigenvectors, $\{x_1\}, \{x_2\}, \ldots, \{x_f\}; \{\bar{\bar{x}}_1\}, \{\bar{\bar{x}}_2\}, \ldots, \{\bar{\bar{x}}_f\}$. Therefore, each new vector pair $\{v_{i+1}\}, \{\bar{\bar{v}}_{i+1}\}$ obtained from either the algorithm or the pseudo-random number generator (see previous Section) is denoted as $\{v_{i+1}^{(0)}\}, \{\bar{\bar{v}}_{i+1}^{(0)}\}$ and reorthogonalized with respect to all the previously obtained vectors. This is accomplished by iterating,

$$\{v_{i+1}^{(k)}\} = \{v_{i+1}^{(k-1)}\} - \sum_{j=1}^{i} [\{\bar{\bar{v}}_j\}^T\{v_{i+1}^{(k-1)}\}]\{v_j\} - \sum_{j=1}^{f} [\{\bar{\bar{x}}_j\}^T\{v_{i+1}^{(k-1)}\}]\{x_j\}, \tag{202}$$

$$\{\bar{\bar{v}}_{i+1}^{(k)}\} = \{\bar{\bar{v}}_{i+1}^{(k-1)}\} - \sum_{j=1}^{i} [\{v_j\}^T\{\bar{\bar{v}}_{i+1}^{(k-1)}\}]\{\bar{\bar{v}}_j\} - \sum_{j=1}^{f} [\{x_j\}^T\{\bar{\bar{v}}_{i+1}^{(k-1)}\}]\{\bar{\bar{x}}_j\}, \tag{203}$$

where $k = 1,2,\ldots,s$ until an acceptable vector pair $\{v_{i+1}^{(s)}\}, \{\bar{\bar{v}}_{i+1}^{(s)}\}$ is found which satisfies the orthogonality criteria

$$\max_{1\le j\le i} |\{\bar{\bar{v}}_j\}^T\{v_{i+1}^{(s)}\}| \le 10^{2-t}, \tag{204}$$

$$\max_{1\le j\le i} |\{v_j\}^T\{\bar{\bar{v}}_{i+1}^{(s)}\}| \le 10^{2-t}, \tag{205}$$

$$\max_{1\le j\le f} |\{\bar{\bar{x}}_j\}^T\{v_{i+1}^{(s)}\}| \le 10^{2-t}, \tag{206}$$

$$\max_{1\le j\le f} |\{x_j\}^T\{\bar{\bar{v}}_{i+1}^{(s)}\}| \le 10^{2-t}, \tag{207}$$

where $s \le 10$, and t is the number of digits carried in the computations. If, for some value of i, the above criteria are not satisfied, then it is assumed that a new pair of trial vectors cannot be generated and a reduced eigenvalue problem of order $m = i$ is solved as discussed in Section 10.6.4.3.

If the above criteria are met, then the resulting vectors are biorthonormalized and set equal to the new trial vector, i.e.,

$$\{v_{i+1}\} = \frac{1}{[\{\bar{\bar{v}}^{(s)}_{i+1}\}^T\{v^{(s)}_{i+1}\}]^{1/2}}\{v^{(s)}_{i+1}\}, \tag{208}$$

$$\{\bar{\bar{v}}_{i+1}\} = \frac{1}{[\{\bar{\bar{v}}^{(s)}_{i+1}\}^T\{v^{(s)}_{i+1}\}]^{1/2}}\{\bar{\bar{v}}^{(s)}_{i+1}\} \tag{209}$$

These new vectors are used to compute the next off-diagonal term in the reduced tridiagonal matrix from the formula

$$d_{i+1} = \{\bar{\bar{v}}_i\}^T[A]\{v_{i+1}\}. \tag{210}$$

However, if

$$|d_{i+1}| \le 10^{-t/2}\,|a_{i,i}|, \tag{211}$$

it is probable that either $\{v_{i+1}\}$ or $\{\bar{\bar{v}}_{i+1}\}$ is a null vector. In this event, a reduced eigenvalue problem of order $m = i$ is solved, as above.

If the criteria given by Equations 204 through 207 and 211 are both met, then the new, normalized vectors are used to continue the reduction algorithm.

10.6.4.6 Error Estimates for the Computed Eigenvalues

Following a development similar to that of Section 10.6.2.7 for real eigenvalue analysis, it can be shown that

$$||\Lambda_i| - |\bar{\Lambda}_i|| \lesssim |d_{m+1}\, y_{mi}|. \tag{212}$$

The above relation shows that the absolute value of the difference between the computed and true eigenvalue magnitudes is proportional to the magnitude of d_{m+1} (which is the next off-diagonal term that would be generated had the reduced tridiagonal matrix [H] been increased from order m to order m+1) and y_{mi}, which is the last term in the reduced-system eigenvector associated with $\bar{\Lambda}_i$.

Converting Equation 212 to physical eigenvalue form, using Equations 163, 165 and 166, yields

$$\left| \frac{|\bar{p}_i - \lambda_o|}{|p_i - \lambda_o|} - 1 \right| \lesssim \frac{|d_{m+1} y_{mi}|}{|\bar{\Lambda}_i|} \; ; \quad [B] \neq [0], \tag{213}$$

$$\left| \frac{|\bar{p}_i^{\,2} - \lambda_o^{\,2}|}{|p_i^{\,2} - \lambda_o^{\,2}|} - 1 \right| \lesssim \frac{|d_{m+1} y_{mi}|}{|\bar{\Lambda}_i|} \; ; \quad [B] = [0]. \tag{214}$$

The use of the above error bound estimates as criteria for acceptable eigensolutions is described in Section 10.6.5.

10.6.5 Summary of Computational Procedures and Flow Charts - Complex Eigenvalue Analysis

Flow diagrams illustrating the computational procedures are shown in Figures 3 and 4. The details of each block are summarized below.

1. Establish Tentative Reduced Problem Size (see Section 10.6.4.3)

Calculate a tentative size, m, of the reduced eigenproblem from

$$m = \min\,[(2\bar{q}+10),(2n-f)]; \quad [B] \neq [0] \tag{215}$$

$$= \min\,[(2\bar{q}+10),\ (n-f)]; \quad [B] = [0], \tag{216}$$

where n is the order of the [K], [B] and [M] matrices, $\bar{q}$, is the number of user-requested eigensolutions, and f is the total number of previously extracted eigensolutions calculated for earlier shift points (λ_o or λ_o^2).

2. Construct Factors of Dynamic Matrix

(a) Set

$$[\bar{D}] = [K+\lambda_o B+\lambda_o^2 M]; \quad [B] \neq [0] \tag{217}$$

$$= [K+\lambda_o^2 M] \quad ; \quad [B] = [0]. \tag{218}$$

(b) Decompose the above dynamic matrix:

$$[\bar{D}] = [L][U], \tag{219}$$

where [L] is unit lower triangular and [U] is upper triangular, using complex arithmetic (subroutine CDC0MP).

Save the triangular factors for later use in premultiplication operations involving the eigenmatrix, [A], and its transpose. If the decomposition fails, the shift value is increased by $\Delta = .02(1+i)$ and a new decomposition is attempted. If this second decomposition is also unsuccessful, then the program assumes that the problem contains unremovable singularities in the neighborhood of the current shift point and proceeds to the next shift point, if any.

3. Execute Complex Tridiagonal Reduction Algorithm (see flow diagram for this block, Figure 4)

3.1 Initialize the Recurrence Algorithm (see Section 10.6.4.2)

Initialize the vector index to $i=0$ and set

$$\{v_o\} = \{\bar{\bar{v}}_o\} = \{0\}. \tag{220}$$

3.2 Generate a Pair of Starting or Restart Vectors and Set $d_{i+1} = 0.0$ (see Section 10.6.4.4)

(a) Construct right and left vectors $\{w_r\}$ and $\{\bar{\bar{w}}_r\}$ using a pseudo-random number generator. The vectors contain $2n$ elements for $[B]\neq[0]$ and n elements for $[B]=[0]$.

(b) Sweep-out components corresponding to $\bar{\Lambda} = 0$:

$$\{w_{i+1}\} = [A]\{w_r\}, \tag{221}$$

$$\{\bar{\bar{w}}_{i+1}\} = [A]^T\{\bar{\bar{w}}_r\}. \tag{222}$$

(c) Normalize the above vectors:

$$\{v_{i+1}^{(0)}\} = \frac{1}{[\{\bar{\bar{w}}_{i+1}\}^T\{w_{i+1}\}]^{1/2}}\{w_{i+1}\}, \tag{223}$$

$$\{\bar{\bar{v}}_{i+1}^{(0)}\} = \frac{1}{[\{\bar{\bar{w}}_{i+1}\}^T\{w_{i+1}\}]^{1/2}}\{\bar{\bar{w}}_{i+1}\}. \tag{224}$$

(d) Set $d_{i+1} = 0.0$ and proceed to block 3.5.

3.3 Create One Pair of Approximate Trial Vectors and One Diagonal Coefficient (see Section 10.6.4.2)

The recurrence algorithm is:

$$\left.\begin{aligned}
a_{i,i} &= \{\bar{\bar{v}}_i\}^T[A]\{v_i\}, \\
\{w_{i+1}\} &= [A]\{v_i\} - a_{i,i}\{v_i\} - d_i\{v_{i-1}\}, \\
\{\bar{\bar{w}}_{i+1}\} &= [A]^T\{\bar{\bar{v}}_i\} - a_{i,i}\{\bar{\bar{v}}_i\} - d_i\{\bar{\bar{v}}_{i-1}\}, \\
\bar{d}_{i+1} &= [\{\bar{\bar{w}}_{i+1}\}^T\{w_{i+1}\}]^{1/2}, \\
\{v_{i+1}^{(0)}\} &= \frac{1}{\bar{d}_{i+1}}\{w_{i+1}\}, \\
\{\bar{\bar{v}}_{i+1}^{(0)}\} &= \frac{1}{\bar{d}_{i+1}}\{\bar{\bar{w}}_{i+1}\},
\end{aligned}\right\} \tag{225}$$

where the above vectors are approximations to the new right and left trial vectors.

3.4 First Normalization Test (see Section 10.6.4.4)

The test is:

$$|\bar{d}_{i+1}| \geq 10^{-t/2}\,|a_{i,i}|. \tag{226}$$

Pass: Proceed directly to block 3.5.

Fail: Return to block 3.2, generate a new pair of restart vectors for $\{v_{i+1}^{(0)}\}$

and $\{\bar{\bar{v}}_{i+1}^{(0)}\}$, then proceed to block 5.5.

3.5 Iterate to Obtain Orthogonalized Trial Vectors (see Section 10.6.4.5)

Designate $\{x_j\}$, $\{\bar{\bar{x}}_j\}$; $j = 1, 2, \ldots, f$ as previously calculated and stored eigenvector pairs, i.e., calculated earlier by complex FEER for previous shift points in the complex plane. Perform the iterations,

$$\{v_{i+1}^{(k)}\} = \{v_{i+1}^{(k-1)}\} - \sum_{j=1}^{i} [\{\bar{\bar{v}}_j\}^T \{v_{i+1}^{(k-1)}\}]\{v_j\}$$

$$- \sum_{j=1}^{f} [\{\bar{\bar{x}}_j\}^T \{v_{i+1}^{(k-1)}\}]\{x_j\}, \tag{227}$$

$$\{\bar{\bar{v}}_{i+1}^{(k)}\} = \{\bar{\bar{v}}_{i+1}^{(k-1)}\} - \sum_{j=1}^{i} [\{v_j\}^T \{\bar{\bar{v}}_{i+1}^{(k-1)}\}]\{\bar{\bar{v}}_j\}$$

$$- \sum_{j=1}^{f} [\{x_j\}^T \{\bar{\bar{v}}_{i+1}^{(k-1)}\}]\{\bar{\bar{x}}_j\}; \tag{228}$$

$k = 1,2\ldots$, until

$$\left.\begin{aligned}
&\max_{1\le j\le i} |\{\bar{\bar{v}}_j\}^T \{v_{i+1}^{(k)}\}| \le 10^{2-t}, \\
&\max_{i\le j\le i} |\{v_j\}^T \{\bar{\bar{v}}_{i+1}^{(k)}\}| \le 10^{2-t}, \\
&\max_{i\le j\le f} |\{\bar{\bar{x}}_j\}^T \{v_{i+1}^{(k)}\}| \le 10^{2-t}, \\
&\max_{i\le j\le f} |\{x_j\}^T \{\bar{\bar{v}}_{i+1}^{(k)}\}| \le 10^{2-t},
\end{aligned}\right\} \tag{229}$$

or $k = 10$.

If the orthogonality criteria, Equations 229, are satisfied, proceed to block 3.6. Otherwise, set the problem size, m, equal to i and proceed to Exit.

3.6 Normalize the Orthogonalized Trial Vectors (see Section 10.6.4.5)

Compute

$$\{v_{i+1}\} = \frac{1}{[\{\bar{\bar{v}}_{i+1}^{(k)}\}^T\{v_{i+1}^{(k)}\}]^{1/2}}\{v_{i+1}^{(k)}\}, \tag{230}$$

$$\{\bar{\bar{v}}_{i+1}\} = \frac{1}{[\{\bar{\bar{v}}_{i+1}^{(k)}\}^T\{v_{i+1}^{(k)}\}]^{1/2}}\{\bar{\bar{v}}_{i+1}^{(k)}\}. \tag{231}$$

These are the new orthogonalized and normalized trial vectors.

3.7 Second Normalization Test and Creation of Off-Diagonal Coefficient (See Section 10.6.4.5)

(a) Compute the next off-diagonal term of the reduced tridiagonal matrix from

$$d_{i+1} = \{\bar{\bar{v}}_i\}^T[A]\{v_{i+1}\}. \tag{232}$$

(b) Verify whether the following test is met:

$$|d_{i+1}| \geq 10^{-t/2}\,|a_{i,i}|. \tag{233}$$

If it is, set $i = i+1$ and return to block 3.3 for continuation of the recurrence algorithm. If the test fails, set $m=i$ to reduce the problem size and proceed to Exit.

4. Solve Reduced-System Eigenproblem (see Section 10.6.4.2)

(a) The coefficients $a_{11}, a_{22}, \ldots, a_{mm}$ and $d_2, d_3, \ldots, d_m$ computed in block 3 form the diagonal and off-diagonal terms of an m^{th} order symmetric, tridiagonal matrix [H] (the matrix is, however, usually complex rather than real).

(b) The m^{th} order eigenvalue problem

$$[H]\{y\} = \bar{\Lambda}\{y\} \tag{234}$$

is solved for the eigenvalues, $\bar{\Lambda}_i$, and the eigenvectors, $\{y_i\}$, using the Q-R iteration algorithm and eigenvector computational scheme described in connection with the Upper Hessenberg method (Section 10.5).

(c) The reduced system eigenvectors are normalized so that

$$\{y_i\}^T\{y_i\} = 1; \quad i=1,\ m. \tag{235}$$

5. Compute Estimate of Eigenvalue Errors (see Section 10.6.4.6)

(a) Estimates of absolute relative errors in the computed eigenvalues are obtained from

$$\xi_i = \left| \frac{|\bar{p}_i - \lambda_o|}{|p_i - \lambda_o|} - 1 \right| \lesssim \frac{|\bar{d}_{m+1} y_{mi}|}{|\bar{\Lambda}_i|}\ ;\ [B] \neq [0], \tag{236}$$

$$\xi_i = \left| \frac{|\bar{p}_i^{\,2} - \lambda_o^{\,2}|}{|p_i^{\,2} - \lambda_o^{\,2}|} - 1 \right| \lesssim \frac{|\bar{d}_{m+1} y_{mi}|}{|\bar{\Lambda}_i|}\ ;\ [B] = [0], \tag{237}$$

where $\bar{d}_{m+1}$ is the last off-diagonal term computed in block 3.3 and y_{mi} is the last element in the vector $\{y_i\}$. If the physical eigenvalue, $\bar{p}_i$, corresponds to a zero root (e.g., a rigid body mode), the above computational scheme is invalid and therefore bypassed. A zero root is assumed to occur whenever

$$\frac{|\bar{p}_i|}{RMS} < 10^{-t/3}, \tag{238}$$

where,

$$RMS = \frac{1}{m}\,[|\bar{p}_1^{\,2}| + |\bar{p}_2^{\,2}| + \ldots\ldots + |\bar{p}_m^{\,2}|]^{1/2}, \tag{239}$$

and is denoted by setting the error, ξ_i, to an exact zero.

(b) The eigenvalues are listed in order of increasing distance from the shift point, λ_o, to determine whether their associated estimated errors, ξ_i, meet an acceptable relative error tolerance set by the user on the EIGC bulk data card (the default value is .001/n where n is the order of the unreduced problem). The first eigenvalue not meeting the tolerance test, as well as all subsequent eigenvalues further removed from the center of interest, are considered to lack sufficient accuracy and are therefore discarded.

(c) Acceptable eigenvalues obtained in the above manner are reordered according to the magnitude of the imaginary part, with positive values considered as a group ahead of all negative values.

6. Compute Physical Eigenvalues and Eigenvectors and Store (see Section 10.6.4.2)

The mathematical eigenvalues, $\bar{\Lambda}_i$, and eigenvectors, $\{y_i\}$, are converted to physical form as follows:

$$\left.\begin{aligned} \bar{p}_i &= \frac{1}{\bar{\Lambda}_i} + \lambda_o \\ \begin{Bmatrix} u \\ v \end{Bmatrix}_i &= [V]\{y_i\}, \end{aligned}\right\} \quad [B] \neq [0] \qquad (240)$$

$$\left.\begin{aligned} \bar{p}_i &= \left(\frac{1}{\bar{\Lambda}_i} + \lambda_o^2\right)^{1/2}; \quad \mathrm{Im}(\bar{p}_i) > 0 \\ \{u_i\} &= [V]\{y_i\} \end{aligned}\right\} \quad [B] = [0] \qquad (241)$$

where

$$[V] = [\{v_1\},\{v_2\},\ldots\ldots,\{v_m\}]. \qquad (242)$$

The velocity vector $\{v_i\}$ in Equation 240 is discarded.

REFERENCES

1. Guyan, R. J., "Reduction of Stiffness and Mass Matrices," AIAA Journal, Vol. 3, No. 2, Feb. 1965, p. 380.
2. Kaufman, S., and Hall, D. B., "Reduction of Mass and Loading Matriçes," AIAA Journal, Vol. 6, No. 3, March 1968, pp. 550-551.
3. Ramsden, R. N., and Stoker, R. J., "Mass Condensation: A Semi-automatic Method for Reducing the Size of Vibration Problems," Int. J. Numer. Meth. Eng., Vol. 1, 1969, pp. 333-349.
4. Crandall, S. H., Engineering Analysis, McGraw-Hill, New York, 1956, pp. 106-109.
5. Lanczos, C., "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators," J. Res. Nat. Bur. Stand., Vol. 45, 1950, pp. 255-282.
6. Gregory, R. T., "Results Using Lanczos Method for Finding Eigenvalues of Arbitrary Matrices," J. Soc. Ind. and Appl. Math., Vol. 6, 1958, pp. 182-188.
7. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.
8. Ojalvo, I. U., and Newman, M., "Vibration Modes of Large Structures by an Automatic Matrix-Reduction Method," AIAA Journal, Vol. 8, No. 7, July 1970, pp. 1234-1239.
9. Newman, M., and Pipano, A., "Vibration Modes via Program NEWLAS" Chapter C, "Program FEER (Fast Eigenvalue Extraction Routine)," Israel Aircraft Industries TR 4842/6277, June 1971, pp. 7-16.
10. Newman, M., and Pipano, A., "Fast Modal Extraction in NASTRAN via the FEER Computer Program," NASA TM X-2893, September 1973, pp. 485-506.
11. Givens, W., "Numerical Computation of the Characteristic Value of a Real Symmetric Matrix," Oak Ridge National Lab., ORNL-1574, 1954.
12. Householder, A. S., "Unitary Triangularization of a Nonsymmetric Matrix," J. Assoc. Comp. Mach., Vol. 5, 1958, pp. 339-342.
13. Potts, J. S., Newman, M., and Wang, S. L., "HABEAS - A Structural Dynamics Analysis System," Proc. 24th Nat. Conf. of the Assoc. for Comp. Mach., ACM Publ. p-69, 1969, pp. 647-664.
14. Causey, R. L., and Gregory, R. T., "On Lanczos' Algorithm for Tridiagonalizing Matrices," Soc. for Industrial and Applied Math. Review, Vol. 3, 1961, pp. 322-328.
15. Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice-Hall, Inc., 1963, p. 108.

ORIGINAL PAGE IS OF POOR QUALITY

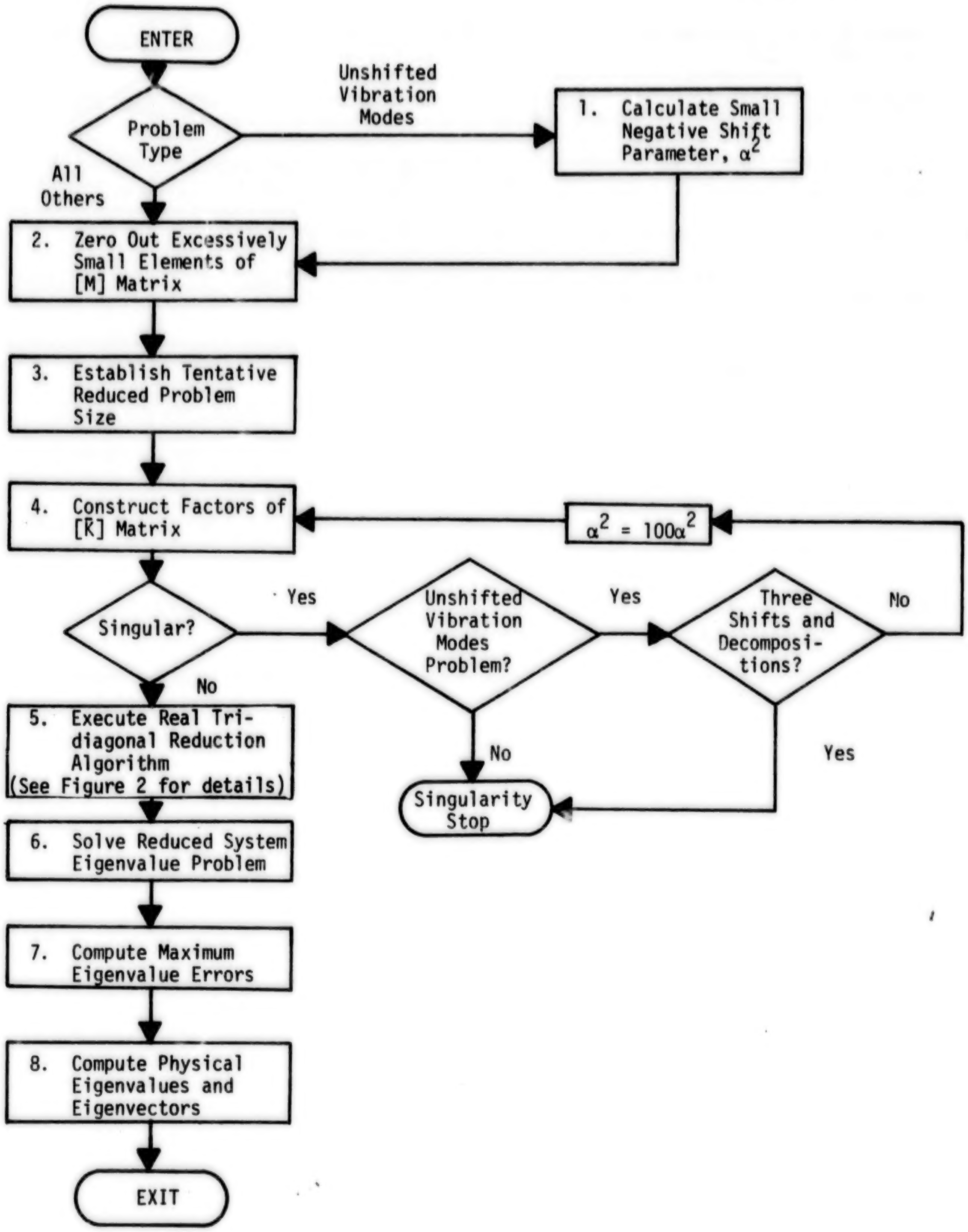

ENTER
Problem Type
Unshifted Vibration Modes
1. Calculate Small Negative Shift Parameter, α²
All Others
2. Zero Out Excessively Small Elements of [M] Matrix
3. Establish Tentative Reduced Problem Size
4. Construct Factors of [K̄] Matrix
α² = 100α²
Singular?
Yes
Unshifted Vibration Modes Problem?
Yes
Three Shifts and Decompositions?
No
No
5. Execute Real Tridiagonal Reduction Algorithm (See Figure 2 for details)
No
Yes
Singularity Stop
6. Solve Reduced System Eigenvalue Problem
7. Compute Maximum Eigenvalue Errors
8. Compute Physical Eigenvalues and Eigenvectors
EXIT

Figure 1. Overall flow diagram for real Tridiagonal Reduction Method.

765

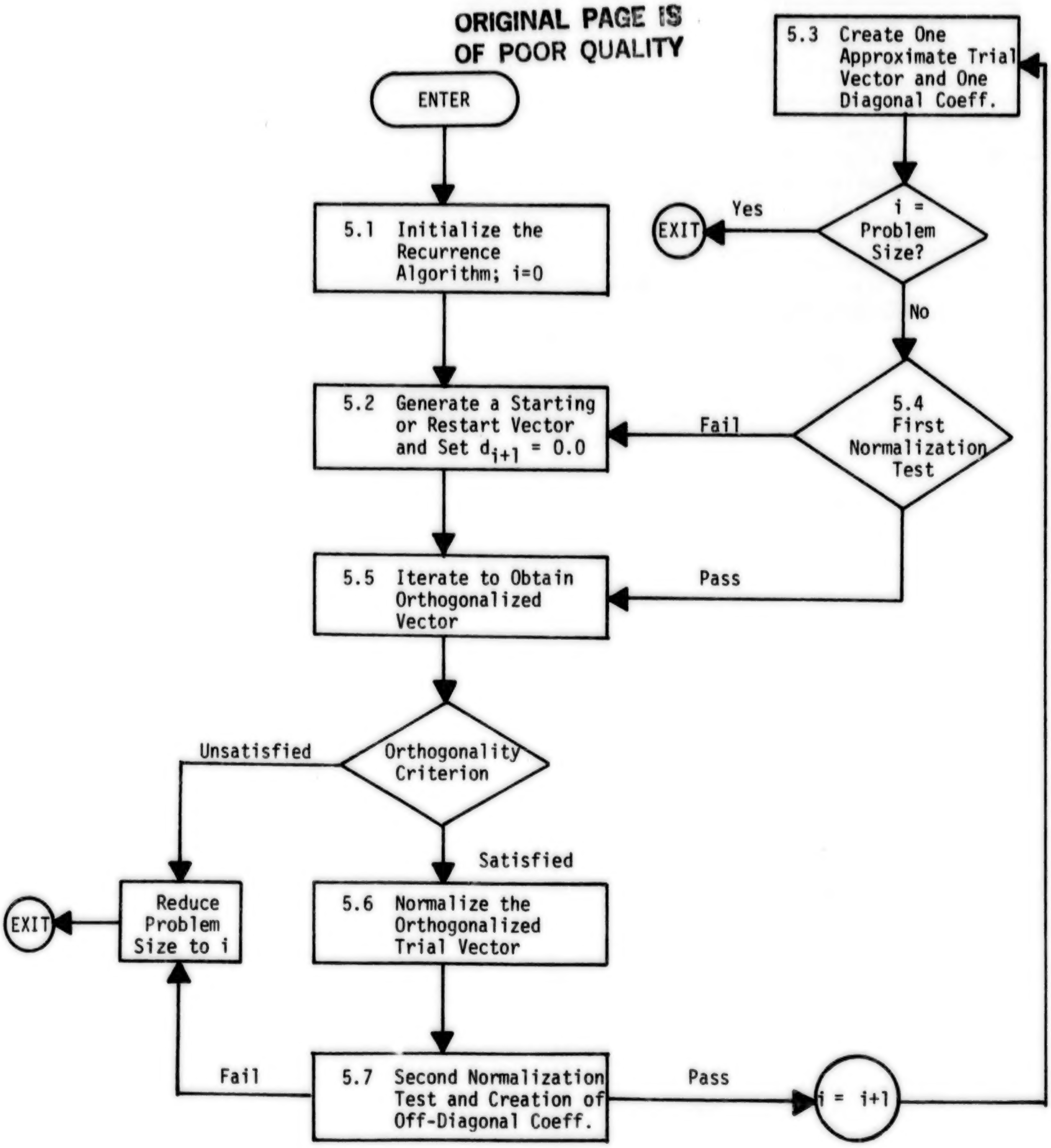

ENTER
5.1 Initialize the Recurrence Algorithm; i=0
5.2 Generate a Starting or Restart Vector and Set d_{i+1} = 0.0
5.5 Iterate to Obtain Orthogonalized Vector
Orthogonality Criterion
Unsatisfied
Satisfied
Reduce Problem Size to i
EXIT
5.6 Normalize the Orthogonalized Trial Vector
5.7 Second Normalization Test and Creation of Off-Diagonal Coeff.
Fail
Pass
i = i+1
5.3 Create One Approximate Trial Vector and One Diagonal Coeff.
i = Problem Size?
Yes
EXIT
No
5.4 First Normalization Test
Fail
Pass

Figure 2. Flow diagram for block 5, Figure 1, Execute Real Tridiagonal Reduction Algorithm.

ORIGINAL PAGE IS
OF POOR QUALITY

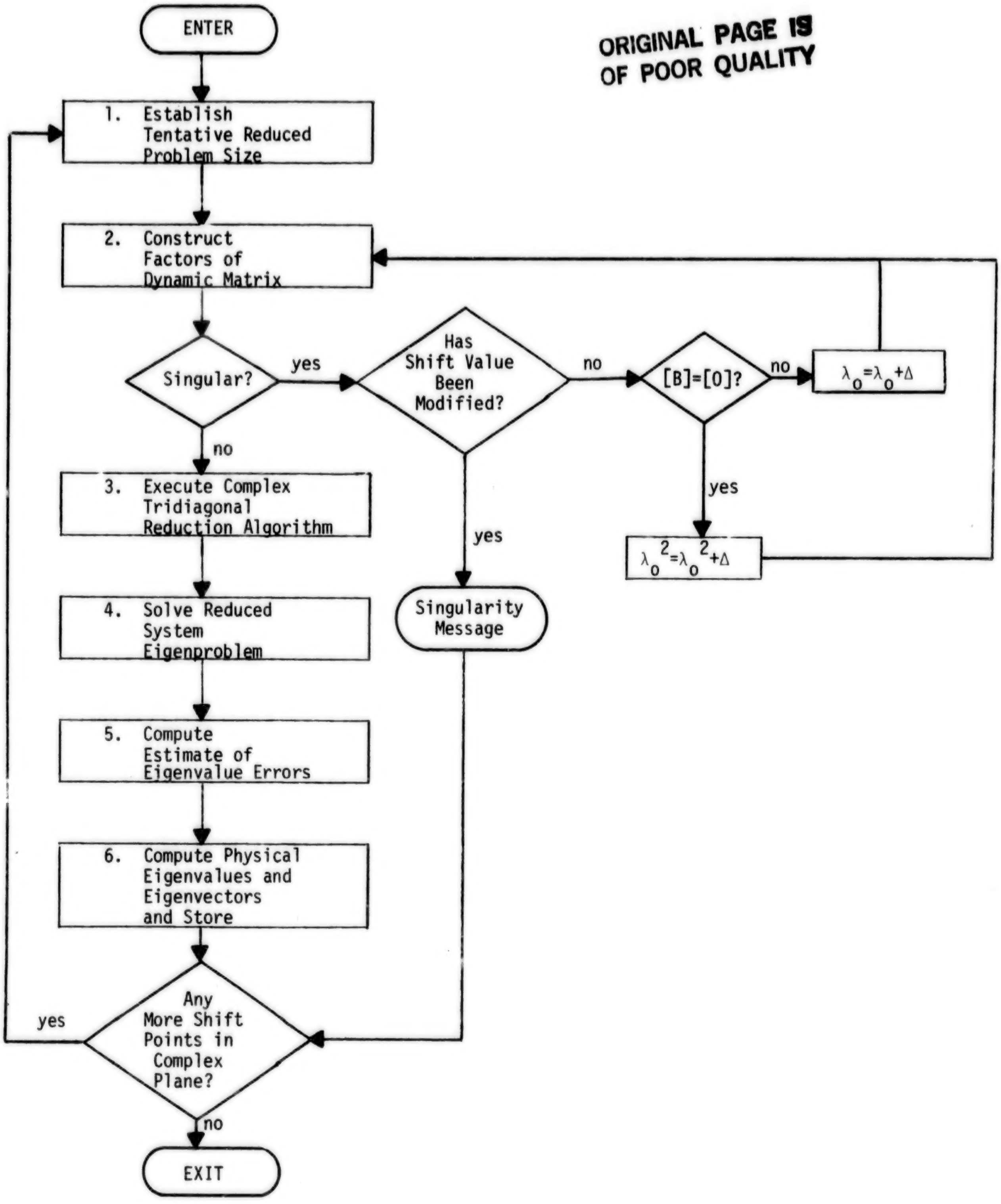

ENTER
1. Establish Tentative Reduced Problem Size
2. Construct Factors of Dynamic Matrix
Singular?
yes
no
Has Shift Value Been Modified?
no
[B]=[O]?
no
$\lambda_o=\lambda_o+\Delta$
yes
$\lambda_o^2=\lambda_o^2+\Delta$
yes
Singularity Message
3. Execute Complex Tridiagonal Reduction Algorithm
4. Solve Reduced System Eigenproblem
5. Compute Estimate of Eigenvalue Errors
6. Compute Physical Eigenvalues and Eigenvectors and Store
Any More Shift Points in Complex Plane?
yes
no
EXIT

Figure 3. Overall flow diagram - complex Tridiagonal Reduction Method

ORIGINAL PAGE IS
OF POOR QUALITY

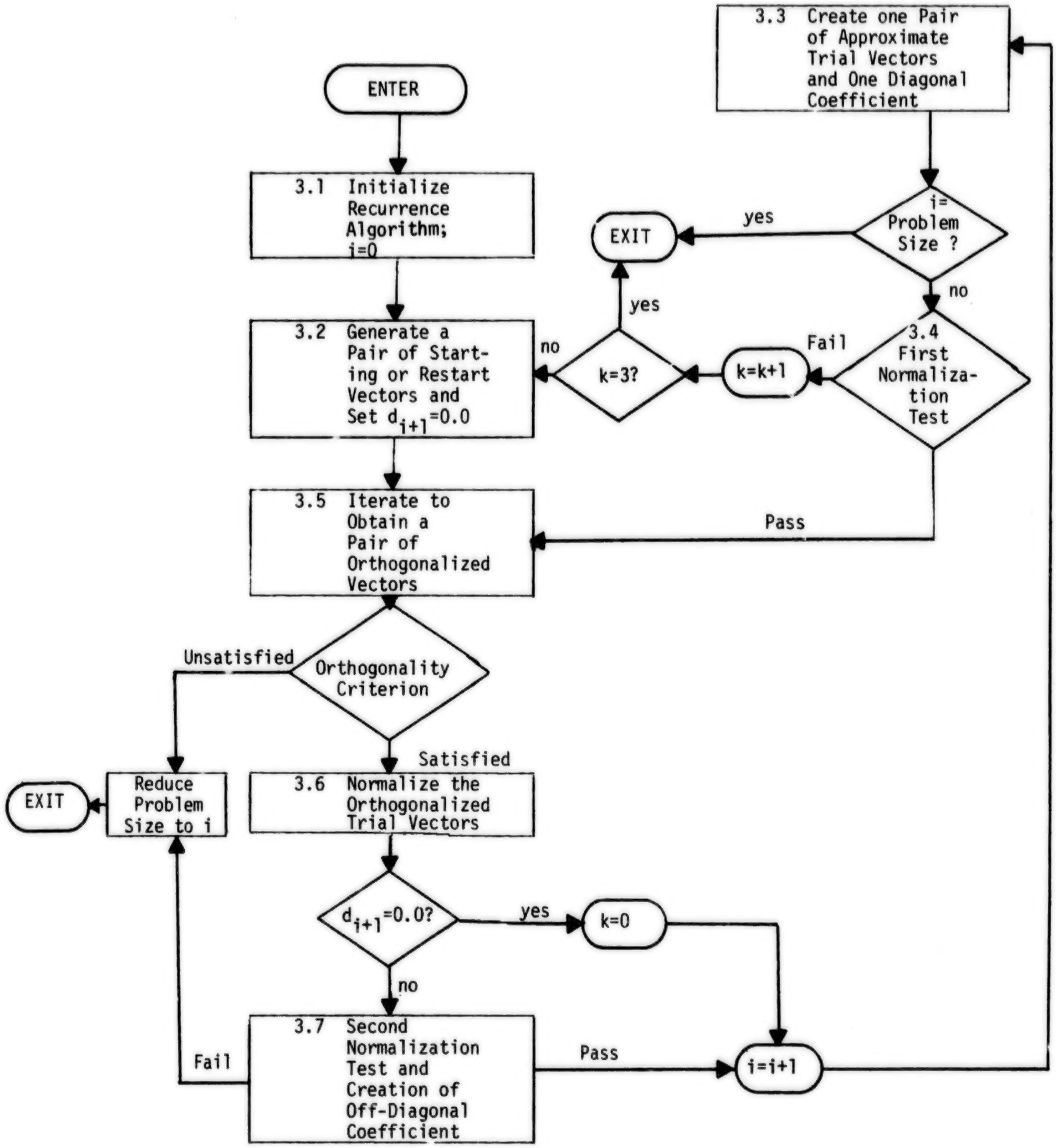

ENTER
3.1 Initialize Recurrence Algorithm; i=0
3.2 Generate a Pair of Starting or Restart Vectors and Set d_{i+1}=0.0
3.3 Create one Pair of Approximate Trial Vectors and One Diagonal Coefficient
i= Problem Size ?
yes
EXIT
no
3.4 First Normalization Test
Fail
k=k+1
k=3?
yes
no
Pass
3.5 Iterate to Obtain a Pair of Orthogonalized Vectors
Orthogonality Criterion
Unsatisfied
Satisfied
Reduce Problem Size to i
EXIT
3.6 Normalize the Orthogonalized Trial Vectors
d_{i+1}=0.0?
yes
k=0
no
3.7 Second Normalization Test and Creation of Off-Diagonal Coefficient
Fail
Pass
i=i+1

Figure 4. Flow diagram for block 3, Execute Complex Tridiagonal Reduction Algorithm.

11. TRANSIENT ANALYSIS

11.1 INTRODUCTION

Figure 1 shows a simplified flow diagram for the transient analysis module. Each of the functional blocks in Figure 1 is discussed in a separate subsection. The equations that are solved by the module may be expressed in terms of physical coordinates, u_d, or in terms of augmented modal coordinates, u_h. Section 9.3 should be consulted for a complete explanation of the assembly of the matrices in the equations of motion. If, in a modal formulation, there is no direct input, i.e., if $[M_{dd}^2] = [B_{dd}^2] = [K_{dd}^2] = 0$, and if there are no nonlinear terms, the equations of motion for the individual modal coordinates are uncoupled and are of the following form:

$$m_i\ddot{\xi}_i + b_i\dot{\xi}_i + k_i\xi_i = P_i \quad . \tag{1}$$

The uncoupled equations are sufficiently simple that they are integrated by analytical rather than strictly numerical methods, see Section 11.5. The coupled equations are solved by a relatively simple numerical integration algorithm that has been developed to meet the requirements of structural analysis, see Section 11.4.

Choosing between the coupled direct solution and the modal coordinate solution involves several considerations. Given a sufficient number of modes, the uncoupled modal solution generally produces more accurate results, but the options are limited. The type of structural damping is restricted. Also, initial conditions other than zero and nonlinear loads may not be applied to the structural grid points. The cost of extracting the modes and the error introduced due to the omission of higher modes must also be considered.

The direct matrix solution to the transient problem is often more costly. Also, smaller time steps are required to retain commensurate accuracy. However, its generality allows complete definition of initial displacements and velocities, as well as damping effects and nonlinear loads. It is most effective in solving problems with impulsive type loads. The accuracy and the stability of the method are discussed in Section 11.4.

The results of the transient analysis module are the displacement vectors, $\{u_d\}$ or $\{u_h\}$, and their first and second time derivatives. They are passed to the Dynamic Data Recovery Module for further processing, see Figure 1 of Section 9.1. The final results, which also include internal forces and stresses, are printed and/or plotted versus time. In addition, for a list of times provided by the user, structural displacements may be superimposed on a plot of the undeformed structure, see Section 13.

PAGE 769 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

TRANSIENT ANALYSIS

A different approach to integrate the equations of motion is used for aeroelastic response problems. Aerodynamic matrices are often known only in the frequency domain, which does not permit direct integration. A Fourier method is used, in which applied loads are transformed to the frequency domain. Frequency responses, computed by the method of Section 12, are then transformed back to the time domain. Section 11.6 presents the theory which is used in NASTRAN for aeroelastic problems.

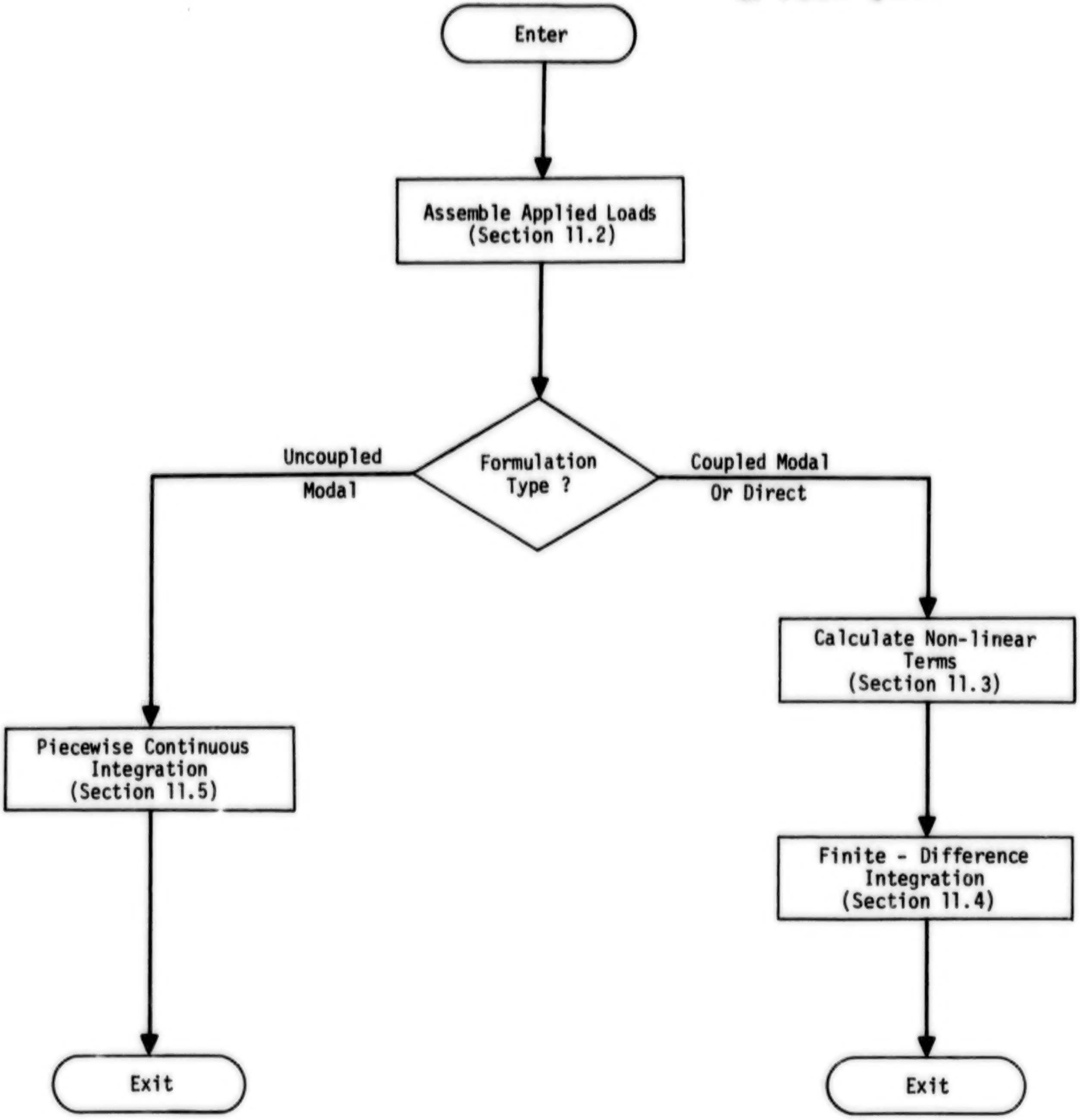

Enter
Assemble Applied Loads
(Section 11.2)
Formulation
Type ?
Uncoupled
Modal
Coupled Modal
Or Direct
Calculate Non-linear
Terms
(Section 11.3)
Piecewise Continuous
Integration
(Section 11.5)
Finite - Difference
Integration
(Section 11.4)
Exit
Exit

Figure 1. Simplified flow diagram for transient analysis module, TRD.

772

11.2 TRANSIENT LOADS

In static analysis a number of automatic load generating subroutines (gravity load, pressure load, temperature induced load, etc.) are employed in order to simplify the user's task of input data preparation. Similar schemes are not made available for transient analysis mainly because of the great variety in the possible sources of dynamic load. Instead the user is provided with flexible input data formats which allow for a varying degree of generality vs. simplicity.

The generation of transient loads can be a formidable task. The most general form of loading is one which has a different time history of load for each point in the structure. Even for a problem of moderate size (say 50 loaded points and 200 time intervals) the data set required to specify a general loading is large (50 x 200 = 10,000 entries). Thus it is essential to provide data formats that accommodate special cases.

In NASTRAN, the applied transient load vector, $\{P_j^c\}$, that is used in a specific subcase is constructed as a combination of component load sets $\{P_j^k\}$,

$$\{P_j^c(t)\} = \sum_k S_{ck}\{P_j^k(t)\} \quad , \tag{1}$$

where S_{ck} is a factor that gives the proportion of component load set k used in combined load set c. The advantage of this system is that it facilitates the examination of different combinations of loads from different sources.

Two separate forms are provided for specifying component load sets. In the first, or general, form

$$\{P_j^k(t)\} = \{A_{jk}\}F_K(t - \tau_{jk}) \quad , \tag{2}$$

where A_{jk} and τ_{jk} are tabulated coefficients that may be different for each loaded degree of freedom (j). (Alternatively, A_{jk} may also refer to the components of a simple static load set vector.) A given table of coefficients (A or τ) may be referenced by more than one component load set. $F_k(t - \tau)$ is a tabulated function of time that is linearly interpolated between entries. The form provided by Equation 2 is particularly useful for loads due to traveling waves. In such problems F_k represents the pressure produced by the wave, A_{jk} is the exposed area associated with the jth degree of freedom, and τ_{jk} is the travel time required for the wave to reach the jth degree of freedom.

In the second, or special, form of transient loading

PAGE 773 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

$$\{P_j{}^k(t)\} = \{A_{jk}\}(\overline{t})^{n_k}\left(e^{\alpha_k\overline{t}}\right)\cos(2\pi f_k\overline{t} + \phi_k), \quad 0 \leqslant \overline{t} \leqslant T_{2k} - T_{1k} \quad , \tag{3}$$

$$\{P_j{}^k(t)\} = 0, \quad 0 > \overline{t} \quad \text{or} \quad \overline{t} > T_{2k} - T_{1k} \quad ,$$

where

$$\overline{t} = t - T_{1k} - \tau_{jk} \quad . \tag{4}$$

The coefficients A_{jk} and τ_{jk} have the same meaning as they do in the general form, Equation 2. The six constants, T_{1k}, T_{2k}, n_k, α_k, f_k and ϕ_k are entered on a separate data card for each load set. Although the appearance of Equation 3 is formidable, its application to a variety of special cases is straightforward. The time constants, T_{1k} and T_{2k}, define the time limits of nonzero load for a point at which the time delay τ_{jk} is zero. A square wave between T_{1k} and T_{2k} is specified by setting $n_k = \alpha_k = f_k = \phi_k = 0$. A sine wave with frequency f starting at $t = T_{1k}$ is specified by setting $n_k = \alpha_k = 0$, $f_k = f$ and $\phi_k = -\pi/2$. A polynomial function, $P = a + bt + ct^2 + \ldots$, is specified by combining a number of load sets with different integral values of n_k and $\alpha_k = \omega_k = \phi_k = 0$. Since blank entries are interpreted as zeroes by the computer, the extra burden on the user due to the generality of Equation 3 is minimal in all of the above examples.

The loads are referred by the user to the displacement set, u_p, of all physical points (structural grid points, u_g, plus extra points, u_e). The reduction to final form consists of applying single and multipoint constraints, the Guyan reduction, and in the case of a modal formulation, the transformation to modal coordinates. Except for the modal transformation, the steps are identical to those for the reduction of static loads to final form described in Section 3.6.2. The loads on extra points are included in their proper sequence in each intermediate load vector.

Since the number of time steps is large (typically of the order of 1000), the reduction procedure would be slow and cumbersome if it were applied directly to the load vector at each time step. A more efficient procedure, if the number of loaded points is fewer than the number of time steps, is to precompute a load transformation matrix, $[A_{dk}]$ or $[A_{hk}]$, such that the final combined load vector is calculated by

$$\{P_d{}^c(t)\} = [A_{dk}]\,\{F_k(t)\} \quad , \tag{5}$$

in the case of a direct formulation, or by

$$\{P_h^{\,c}(t)\} \quad = \quad [A_{hk}]\{F_k(t)\} \qquad (6)$$

in the case of a modal formulation. $\{F_k(t)\}$ is the vector of unique loading functions of time, and the matrices $[A_{dk}]$ and $[A_{hk}]$ define constant factors for each load function and degree of freedom.

The load transformation matrices are calculated as follows (compare with Section 3.6.2):

1. Form a matrix $[A_{pk}]$ with the number of rows, p, equal to the total number of physical points, and the number of columns, k, equal to the number of unique load functions of time. The elements of $[A_{pk}]$ correspond to the factors A_{jk} in Equation 2.

2. Partition $[A_{pk}]$ into $[A_{mk}]$, points eliminated by multipoint constraints, and $[\bar{A}_{nk}]$.

$$[A_{pk}] \quad = \quad \begin{bmatrix} \bar{A}_{nk} \\ \hdashline A_{mk} \end{bmatrix} \; . \qquad (7)$$

3. Apply multipoint constraints.

$$[A_{pk}] \quad = \quad [\bar{A}_{nk}] + [G_m]^T[A_{mk}] \quad . \qquad (8)$$

4. Partition $[A_{nk}]$ into $[A_{sk}]$, points eliminated by single point constraints, and $[A_{fk}]$.

$$[A_{fk}] \quad = \quad \begin{bmatrix} A_{fk} \\ \hdashline A_{sk} \end{bmatrix} \; . \qquad (9)$$

5. Partition $[A_{fk}]$ into $[A_{ok}]$, points eliminated by the Guyan reduction, and $[\bar{A}_{dk}]$.

$$[A_{fk}] \quad = \quad \begin{bmatrix} \bar{A}_{dk} \\ \hdashline A_{ok} \end{bmatrix} \; . \qquad (10)$$

6. Apply the Guyan reduction.

$$[A_{dk}] \quad = \quad [\bar{A}_{dk}] + [G_o]^T[A_{ok}] \quad . \qquad (11)$$

7. In a direct problem formulation, the matrix of load vectors is calculated by the equation

$$[P_{dt}] = [A_{dk}]\,[F_{kt}] \quad , \qquad (12)$$

where each column of $[P_{dt}]$ and $[F_{kt}]$ corresponds to a time step in the integration.

8. For a modal problem formulation only, transform $[A_{dk}]$ to the modal coordinates (see Section 9.3).

$$[A_{hk}] = [\phi_{dh}]^T[A_{dk}] \quad . \qquad (13)$$

9. The matrix of load vectors in the modal coordinates are:

$$[P_{ht}] = [A_{hk}]\,[F_{kt}] \quad , \qquad (14)$$

where each column of $[P_{ht}]$ and $[F_{kt}]$ corresponds to a time step in the integration procedure. The rows of $[P_{ht}]$ correspond to the modal degrees of freedom. The rows of $[F_{kt}]$ correspond to the different loading functions.

The above procedure is used in both structural and heat transfer transient solutions. In structural analysis, each combination of load function input and each time delay, τ, for that function of time, defines a unique term $F_k(t)$. In heat transfer analysis, additional functions of time are defined for the cases of vector heat flux loads which change direction with time.

11.3 NONLINEAR ELEMENTS

Nonlinear effects are treated as an additional applied load vector, {N}, whose elements are functions of either displacements or velocities. {N} is added to the right side of the equations of motion and is treated in a similar but slightly different manner than the applied load vector, {P}, during numerical integration (see Equation 31 of Section 11.4).

In order to eliminate the cumbersome task of reducing nonlinear loads from the u_p displacement set to the u_d or u_h displacement sets, it is required that the points to which the nonlinear loads are applied and the degrees of freedom on which they depend be members of the u_d set, i.e., that they not be points eliminated by constraints. It is further required, for the same reason, that if a modal formulation is used, the points referenced by the nonlinear loads be members of the u_e set (extra points). Otherwise two complete modal transformations (one for displacements and one for loads) would be required at each time step. The means for working around this restriction are explained later on.

NASTRAN at present includes only four different types of nonlinear elements. They are, however, sufficient to generate almost any type of nonlinear relationship when used in conjunction with transfer functions and direct input matrices. It is expected that additional types will be added later as a convenience to the user.

The four nonlinear elements are:

1. Arbitrary Function Generator

$$N_i(t) = S_i F(x_j) \quad , \tag{1}$$

where

$N_i(t)$ is the load applied to u_i,

S_i is an arbitrary (convenience) factor,

$F(x_j)$ is a tabulated function,

and x_j is any (permissible) displacement or velocity component.

The function $F(x_j)$ is interpolated linearly. Several nonlinear elements may reference the same table, in order to avoid duplication of input preparation when several nonlinear elements of the same type are used.

2. Multiplier

$$N_i(t) = S_i x_j \cdot y_k \tag{2}$$

x_j and y_k are any (permissible) pair of displacement or velocity components. They may be the same.

3. Positive Power Function

$$\left.\begin{aligned} N_i(t) &= S_i(x_j)^{\alpha} & x_j &> 0 \quad , \\ &= 0 & x_j &\leqslant 0 \quad . \end{aligned}\right\} \tag{3}$$

4. Negative Power Function

$$\left.\begin{aligned} N_i(t) &= -S_i(-x_j)^{\alpha} & x_j &< 0 \quad , \\ &= 0 & x_j &\geqslant 0 \quad . \end{aligned}\right\} \tag{4}$$

The negative power function is the negative reflection of the positive power function for the negative range of x_j. α may be any positive or negative real number. Taken together, the positive and negative power functions are useful in representing such things as pneumatic springs, hydraulic dampers, and latching mechanisms whose characteristics are frequently represented by power functions.

As an example of the application of the nonlinear elements, consider the Coulomb damper connected between degrees of freedom u_1 and u_2 shown below.

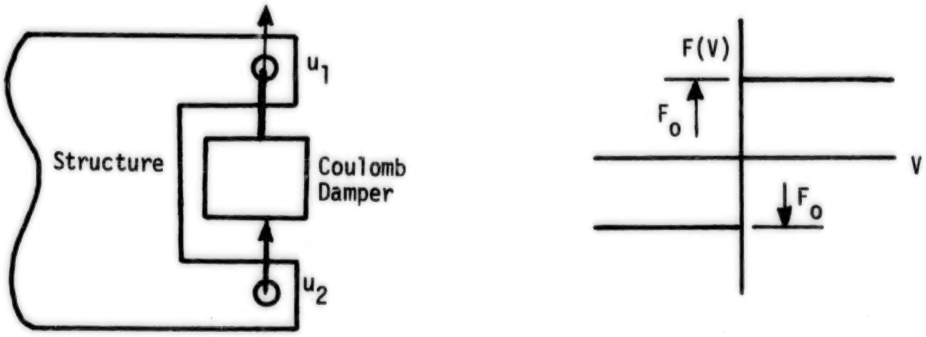


Figure 1. Coulomb damper.

The forces exerted on the degrees of freedom are

$$\left.\begin{aligned} F_1 &= F(V) \quad , \\ F_2 &= -F(V) \quad , \end{aligned}\right\} \qquad (5)$$

where $V = \dot{u}_2 - \dot{u}_1$ and $F(V)$ is the step function of velocity shown above. The arbitrary function generator is used to simulate the Coulomb damper, but if the independent variable is taken to be a displacement, it is first necessary to create an extra point u_{e_1} such that

$$u_{e_1} = \dot{u}_2 - \dot{u}_1 = pu_2 - pu_1 \quad . \qquad (6)$$

This may be done by means of a transfer function (see Equation 10 of Section 9.3.2). We may then declare a pair of nonlinear loads

$$\left.\begin{aligned} N_1 &= F(u_{e_1}) \quad , \\ N_2 &= -F(u_{e_1}) \quad , \end{aligned}\right\} \qquad (7)$$

applied respectively to u_1 and u_2.

Alternately, if the independent variable is taken to be a velocity, the nonlinear loads can be applied to u_1 and u_2 without the need for creating an extra point. Thus

$$\left.\begin{aligned} N_1 &= F(\dot{u}_2) - F(\dot{u}_1) \quad , \\ N_2 &= F(\dot{u}_1) - F(\dot{u}_2) \quad . \end{aligned}\right\} \qquad (7a)$$

In order to avoid difficulty with numerical integration, the step in the force-velocity curve shown in Figure 1 should be replaced by a ramp as shown below.

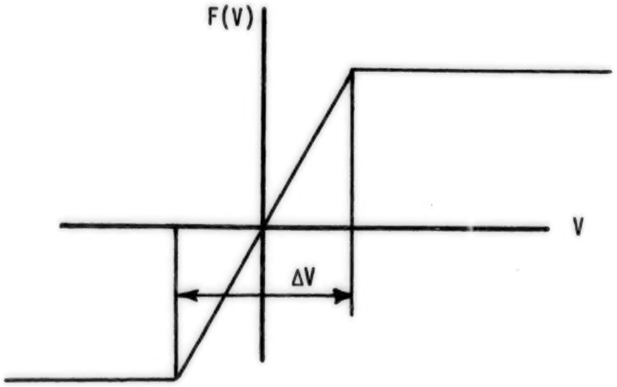


Figure 2. Recommended force-velocity curve for coulomb damper.

The slope of the ramp should be large enough that, when the velocity changes sign, the value of the velocity will lie within the interval ΔV for at least a few time steps in the numerical integration.

If a modal formulation has been used, the nonlinear loads may not be applied directly to structural gridpoints. In this case a second extra point, u_{e_2}, is created and the nonlinear load is applied to it;

$$N_{e_2} = F(u_{e_1}) \quad . \tag{8}$$

In addition, terms are added to the direct input matrix, $[K_{pp}^2]$, in order to provide u_{e_2} with a unit spring restraint (so that it will be numerically equal to N_{e_2}) and to produce forces on u_1 (and u_2) that are equal (and opposite) to u_{e_2}. The added terms to the equations of motion are

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_{e_2} \\ u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} N_{e_2} \\ 0 \\ 0 \end{Bmatrix} \quad . \tag{9}$$

The Coulomb damper may also be used to assist in the dynamic representation of plastic deformation as shown in Figure 3 below.

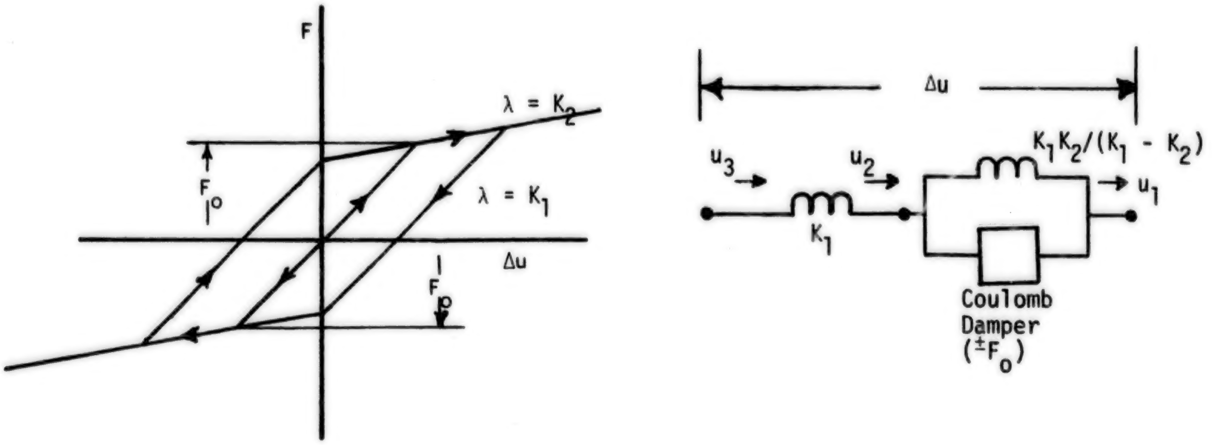


Figure 3. Plastic deformation element.

NONLINEAR ELEMENTS

The springs shown in Figure 3 may be represented by scalar elements. The Coulomb damper is represented in the manner described previously. The composite element exhibits the hysteresis illustrated in Figure 3, which is a characteristic of plastic deformation. Additional break points in the force-displacement curve can be produced by adding more springs and more Coulomb dampers.

11.4 INTEGRATION OF COUPLED EQUATIONS

11.4.1 Solution Technique and Procedure

The price paid for high efficiency in the numerical integration of differential equations is a tendency toward instability. Frequently, a choice exists between an efficient algorithm that is unstable for large time steps and an inefficient algorithm that is excessively stable and relatively inaccurate.

In a structural dynamics problem, the stability limit of the integration algorithm may be expressed as the ratio of the maximum permissible time step to the period of the highest vibration mode of the system. For very large systems a limitation of this kind is intolerable because the period of the highest mode of the system is generally not known and is, in fact, zero for the very practical case when the mass matrix is singular. Thus, every effort should be made to provide an integration algorithm that is stable for the widest possible spectrum of practical problems without sacrificing either accuracy or efficiency.

An integration algorithm will be described that satisfies these requirements. It is a particular form of the Newmark Beta Method (Reference 1). The stability of the algorithm will be examined and compared with the stability of other algorithms in the class to which it belongs.

The differential equations of a linear structural problem may be written in the general matrix form

$$[p^2M + pB + K]\{u\} = \{P\} \quad , \tag{1}$$

where $p = \frac{d}{dt}$.

Numerical integration is achieved by replacing p and p^2 by finite difference operators.

$$\begin{aligned} p &= f_1(E) \quad , \\ p^2 &= f_2(E) \quad , \end{aligned} \tag{2}$$

where the shifting operator, E, is defined by

$$u_{n+1} = Eu_n \quad . \tag{3}$$

u_n is the value of u at $t = t_n$.

In examining the stability of a numerical integration procedure as applied to Equation 1, it

PAGE 783 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

is convenient to transform {u} into the normal coordinates {ξ} by means of the matrix of eigenvectors, [Φ]:

$$\{u\} = [\Phi]\{\xi\} \quad . \tag{4}$$

The resulting equations for {ξ} are uncoupled and their stability may be examined one at a time. The actual integration algorithm is applied to {u} rather than {ξ}. Before making the transformation, we must be sure that the transformation matrix [Φ] is independent of the way in which p is related to E in order to ensure that the results of the stability analysis apply to the actual algorithm. If [B] is a linear combination of [M] and [K], i.e., if

$$[B] = a[M] + b[K] \quad , \tag{5}$$

where a and b are constants, the damping is said to be "uniform". For a condition of uniform damping the eigenvalue problem is

$$[(p^2 + ap)M + (1 + bp)K]\{u\} = 0 \quad , \tag{6}$$

or, dividing by 1 + bp,

$$[\lambda M + K]\{u\} = 0 \quad , \tag{7}$$

where

$$\lambda = \frac{p^2 + ab}{1 + bp} = \frac{\left(f_2(E)\right)^2 + a\, f_1(E)}{1 + b\, f_1(E)} \quad . \tag{8}$$

The eigenvalues, λ_i, of Equation 7 are related by Equation 8 to the corresponding eigenvalues, E_i, of the shift operator. The eigenvectors of Equation 7 depend only on the values of the elements of [M] and [K]. They are, therefore, independent of the functional relationships between λ, p, p^2 and E.

For the more general case of nonuniform damping, the eigenvectors are not independent of the functional relationships between p, p^2 and E. For such cases the stability theory to be presented can only be regarded as approximate.

The question of stability has, by the above maneuver, been reduced to the examination of a single second order differential equation that is representative of the uncoupled equations for the normal coordinates, namely

$$(Mp^2 + Bp + K)\xi = P \quad . \tag{9}$$

In performing numerical integration of this equation by whatever scheme, the differential operator, $Mp^2 + Bp + K$, is replaced by a function of the shifting operator, $f(E)$, so that the stability equation becomes

$$f(E) = 0 \quad , \tag{10}$$

which may have a number of roots, E_1, E_2, E_3, etc. The condition for stability of the integration scheme is that the amplitude of the homogeneous solution not increase as time increases, which requires that

$$|E_i| \leqslant 1 \quad \text{for} \quad i = 1, 2, 3. \ . \ . \ , \tag{11}$$

for each of the normal coordinate equations. This is required because, from Equation 3

$$\left| \frac{u_{n+1}}{u_n} \right| = |E| \quad . \tag{12}$$

For all numerical integration procedures $f(E)$ is, or may be reduced to, a polynomial. Consider, for example, the case in which we let

$$p\xi = \frac{1}{\Delta t}\left(\xi_{n+1} - \xi_n\right) = \frac{1}{\Delta t}\left(E - 1\right)\xi_n \quad , \tag{13}$$

$$p^2\xi = \frac{1}{\Delta t^2}\left(\xi_{n+2} - 2\xi_{n+1} + \xi_n\right) = \frac{1}{\Delta t^2}\left(E^2 - 2E + 1\right)\xi_n \quad . \tag{14}$$

Then the stability equation is

$$Mp^2 + Bp + K = \frac{M}{\Delta t^2}\left(E^2 - 2E + 1\right) + \frac{B}{\Delta t}\left(E - 1\right) + K = 0 \quad , \tag{15}$$

which is a quadratic equation in E.

The cases in which $f(E)$ is a second degree polynomial are of special interest because they represent the simplest integration schemes that can be applied to a second order differential equation and also because they permit a thorough algebraic examination of the question of stability. Let the quadratic equation be written in the form,

$$E^2 + 2bE + c = 0 \quad , \qquad (16)$$

where the constant coefficients b and c are real functions of the physical parameters of the problem.

The roots of the stability equation are

$$E_{1,2} = -b \pm \sqrt{b^2 - c} \quad . \qquad (17)$$

For $c \geqslant b^2$, $|E_{1,2}| = c$, so that for this condition the system is stable provided that $c < 1$. For $c < b^2$, the roots of Equation 17 are both real, so that

$$|E_{1,2}| = |b| \pm \sqrt{b^2 - c} \quad . \qquad (18)$$

Stability limits may be obtained by substituting the limiting value $|E_{1,2}| = 1$ into Equation 18 and solving for c in terms of b. The result is

$$c = -1 \pm 2b \quad , \qquad (19)$$

which represents a pair of straight lines in the b, c plane.

The stability limits that have been found bound a region of the b, c plane within which both solutions to the quadratic equation are stable. The region is depicted in Figure 1, below.

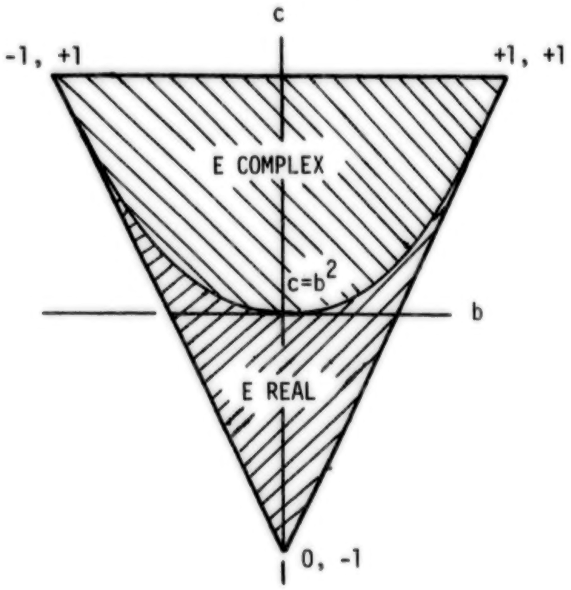


Figure 1. Stability triangle.

As an example of the application of the above analysis, consider Equation 15 with the damping coefficient, B, set equal to zero. For this condition Equation 15 may be written as

$$E^2 - 2E + 1 + \Delta\phi^2 = 0 \quad , \tag{20}$$

where $\Delta\phi = \Delta t \sqrt{\frac{K}{M}}$ is the phase change associated with the time step Δt. The stability parameters are $b = -1$, $c = 1 + \Delta\phi^2$. The point (b,c) is outside the stability triangle for any value of $\Delta\phi$ except zero and the numerical integration will yield an unstable solution.

As a second, and more important example, consider the following central difference equivalent of Equation 9 wherein (a) is an arbitrary coefficient, to be determined.

$$\frac{M}{\Delta t^2}\left(\xi_{n+2} - 2\xi_{n+1} + \xi_n\right) + \frac{B}{2\Delta t}\left(\xi_{n+2} - \xi_n\right) + K\left(a\xi_{n+2} + (1 - 2a)\xi_{n+1} + a\xi_n\right) = P_{n+1} \quad . \tag{21}$$

Note that the expression is symmetrical with respect to the n + 1st time step, a condition that will result in greater accuracy than an unsymmetrical expression. The coefficient (a) will be determined by stability considerations. Normalize Equation 21 by multiplying by $\Delta t^2/M$, which results in the following stability equation:

$$E^2 - 2E + 1 + \frac{\tau}{2}\left(E^2 - 1\right) + \Delta\phi^2\left(aE^2 + (1 - 2a)E + a\right) = 0 \quad , \tag{22}$$

where $\tau = \Delta tB/M$ is a dimensionless time constant and $\Delta\phi$ has the same meaning as before.

The stability parameters are

$$\left.\begin{aligned} b &= \frac{-1 + \frac{1}{2}\left(1 - 2a\right)\Delta\phi^2}{1 + \frac{\tau}{2} + a\Delta\phi^2} \quad , \\ c &= \frac{1 - \frac{\tau}{2} + a\Delta\phi^2}{1 + \frac{\tau}{2} + a\Delta\phi^2} \quad . \end{aligned}\right\} \tag{23}$$

The case $\tau = 0$ is of special interest because it implies no structural damping in which case the original differential equation from which Equation 21 was derived is marginally stable for all values of $\Delta\phi$. For this case

$$\left.\begin{aligned} b &= \frac{-1+\frac{1}{2}\left(1-2a\right)\Delta\phi^2}{1+a\Delta\phi^2} \quad, \\ c &= 1 \quad, \end{aligned}\right\} \tag{24}$$

so that, as $\Delta\phi$ increases from zero, the point in the b, c plane starts at the upper left corner of the stability triangle and moves along the upper edge of the triangle. We desire that the system be marginally stable for all values of $\Delta\phi$. In Equation 24 let b attain its limiting value, +1 as $\Delta\phi \to \infty$. Then for the system to be stable

$$1 \geqslant \frac{1-2a}{2a} \quad, \tag{25}$$

or

$$a \geqslant \frac{1}{4} \quad. \tag{26}$$

This is an interesting result because it shows that unconditional stability (instability independent of size of time step) can be achieved with Equation 21 for certain values of (a). In the usual finite difference approximation, (a) is set equal to zero which, of course, gives unstable results for sufficiently large $\Delta\phi$.

It remains to show that the system point remains within the stability triangle for all positive values of τ and all values of $\Delta\phi$, provided that Equation 26 is satisfied. The system point remains below the upper edge of the triangle, c = 1, for positive τ, as may be seen from Equation 23. The condition for crossing either of the lower edges is, from Equation 19

$$c + 1 \pm 2b = 0 \quad, \tag{27}$$

which becomes, after substituting for b and c from Equation 23

$$\frac{1-\frac{\tau}{2}+a(\Delta\phi)^2}{1+\frac{\tau}{2}+a(\Delta\phi)^2} + 1 \pm \left[\frac{-2+(1-2a)(\Delta\phi)^2}{1+\frac{\tau}{2}+a\Delta\phi^2}\right] = 0 \quad. \tag{28}$$

The denominator may be cleared since it is always positive for $\tau > 0$ and $a > 0$, with the result

$$\Delta\phi^2 = 0 \quad, \tag{29}$$

when the plus sign is chosen, and

$$4 + (4a - 1)\Delta\phi^2 = 0 \quad , \tag{30}$$

when the minus sign is chosen. Neither condition can be satisfied for a finite $\Delta\phi$ provided that $a \geq 1/4$. Furthermore, b and c are, from the form of Equation 23, continuous functions of τ and $\Delta\phi$ for $a \geq 0$. The system point never crosses the boundaries of the stability triangle, and the system is stable for any $\Delta\phi$ provided that $\tau \geq 0$ and $a \geq 1/4$. Thus, if the original system of differential equations is stable, the substitute finite difference system will also be stable. Furthermore, if one of the roots of the original system is neutrally stable, ($\tau_i = 0$), the corresponding root of the substitute system is also neutrally stable.

In the integration algorithm provided with NASTRAN (a) is chosen equal to 1/3, which is somewhat larger than the stability limit, and which provides a margin of stability for more general problems (e.g., those in which the damping is nonuniform or in which nonlinear terms occur) than have been considered in the derivation. The difference equation, in terms of physical coordinates, u, based upon Equation 21 with (a) set equal to 1/3, is:

$$\left[\frac{1}{\Delta t^2} M + \frac{1}{2\Delta t} B + \frac{1}{3} K\right]\{u_{n+2}\} = \frac{1}{3}\{P_{n+2} + P_{n+1} + P_n\} + \{N_{n+1}\} + \left[\frac{2}{\Delta t^2} M - \frac{1}{3} K\right]\{u_{n+1}\}$$

$$+ \left[\frac{-1}{\Delta t^2} M + \frac{1}{2\Delta t} B - \frac{1}{3} K\right]\{u_n\} \quad . \tag{31}$$

$\{N_{n+1}\}$ is the nonlinear load vector described in Section 11.3. Note that the load vector $\{P\}$ is averaged over three adjacent time steps in the same manner that [K] is averaged. This is done in order to provide statically correct solutions for massless degrees of freedom.

As long as the time step, Δt, remains constant, the matrix coefficients of $\{u_{n+2}\}$, $\{u_{n+1}\}$ and $\{u_n\}$ do not change with time and they may be precomputed. The coefficient of $\{u_{n+2}\}$ is decomposed in a preliminary step into its triangular factors $[L_2]$ and $[U_2]$. The solution algorithm consists of the forward pass

$$[L_2]\{y_{n+2}\} = \frac{1}{3}\{P_{n+2} + P_{n+1} + P_n\} + \{N_{n+1}\} + [A_1]\{u_{n+1}\} + [A_0]\{u_n\} \; , \tag{32}$$

and the backward pass

$$[U_2]\{u_{n+2}\} = \{y_{n+2}\} \quad , \tag{33}$$

where

$$[A_1] = \left[\frac{2}{\Delta t^2} M - \frac{1}{3} K\right] \quad , \tag{34}$$

$$[A_0] = \left[\frac{-1}{\Delta t^2} M + \frac{1}{2\Delta t} B - \frac{1}{3} K\right] \quad . \tag{35}$$

The number of scalar multiplications required at each time step is approximately equal to 4Nb where N is the number of equations and b is the bandwidth. A forward integration scheme in which the coefficient of $\{u_{n+2}\}$ is $[M]/\Delta t^2$ where [M] is diagonal and nonsingular, requires about one-half as many multiplications. The latter procedure, however, has stability and accuracy difficulties which can be overcome by reducing the time step to a very small value.

A great advantage of the unconditionally stable integration procedure is that it permits large phase steps in the higher modes without incurring instability. The only limitation on the time step is provided by accuracy considerations. It should be noted in this regard that excessively small time steps produce large round-off errors while large time steps produce large finite difference errors.

A measure of finite difference error in the numerical integration is the discrepancy between the eigenvalues of the original system of differential equations and the eigenvalues of the substitute finite difference system. Of equal importance is the discrepancy in the amplitude of the transient response. Numerical studies indicate that errors in amplitude and errors in frequency have the same order of magnitude. A conventional error analysis (see, for example, Reference 2) of the integration algorithm has been made in which it is assumed that the phase step, $\Delta\phi$, is small compared to unity. Under these conditions the error in frequency (for a = 1/3) is given by

$$\frac{\omega_f}{\omega_e} = 1 - \frac{1}{8}\Delta\phi^2 + --- \quad , \tag{36}$$

where ω_f is the frequency for the finite difference system and ω_e is the exact frequency. If $\Delta\phi$ is 1/4 radian, the error in frequency is a little less than 1%. The error in damping is given by

$$\frac{\zeta_f}{\zeta_e} = 1 - \frac{1}{3}\Delta\phi^2 + --- \quad . \tag{37}$$

One percent error in damping is obtained for $\Delta\phi$ equal to about 1/6 radian. Note that the error in damping is proportional rather than additive as in the case of bootstrap integration procedures in which only the mass matrix is inverted.

In order to illustrate the error in the amplitude of response to transient excitation, consider a simple undamped oscillator with prescribed initial conditions

$$\left.\begin{aligned} m\ddot{u} + ku &= 0 \quad , \\ u(o) &= 1 \quad , \\ \dot{u}(o) &= 0 \quad . \end{aligned}\right\} \tag{38}$$

The correct solution is

$$u = \cos\left(\sqrt{\frac{k}{m}}\, t\right) = \cos\left(\sqrt{\frac{k}{m}}\, n\Delta t\right) \quad . \tag{39}$$

The approximating difference equation is

$$\frac{m}{\Delta t^2}\left(u_{n+2} - 2u_{n+1} + u_n\right) + \frac{k}{3}\left(u_{n+2} + u_{n+1} + u_n\right) = 0 \quad , \tag{40}$$

with initial conditions

$$\begin{aligned} u_0 &= 1 \quad , \\ u_{-1} &= 1 \quad . \end{aligned}$$

The exact solution of the difference equation and the initial conditions, as obtained by the method of undetermined coefficients. is

$$u_n = \cos n\beta + \frac{\cos\beta - 1}{\sin\beta} \sin n\beta \quad , \tag{41}$$

where

$$\cos\beta = \frac{1 - \frac{1}{6}\Delta\phi^2}{1 + \frac{1}{3}\Delta\phi^2} \quad , \tag{42}$$

and the phase step

$$\Delta\phi = \sqrt{\frac{k}{m}}\,\Delta t \quad . \tag{43}$$

For small phase steps, from Equation 36

$$\beta = \omega_f \Delta t = \omega_e \Delta t \frac{\omega_f}{\omega_e} = \Delta\phi \frac{\omega_f}{\omega_e} \simeq \Delta\phi(1 - \frac{1}{8}\Delta\phi^2) \quad . \qquad (44)$$

The ratio of the approximate amplitude of response obtained from Equation 41 to the exact amplitude of response is

$$\frac{A_f}{A_e} = \frac{2 \sin \frac{\beta}{2}}{\sin \beta} \simeq 1 + \frac{1}{8}\Delta\phi^2 \quad . \qquad (45)$$

Thus the error in amplitude has the same value and opposite sign as the error in frequency.

An estimate of the total number of time steps required may be obtained with the aid of Equation 36. For example, assume that a solution is to be obtained for a time interval equal to five times the period of the lowest mode. Assume further that 10% error in frequency and amplitude is permissible for modes with frequencies higher than ten times the frequency of the lowest mode. The permissible phase step relative to the lowest mode is therefore about .09 radian. The total number of time steps is 5 x 2π/.09 = 349.

Initial conditions are specified by the user as the values of physical displacements $\{u_p\}$ and velocities $\{\dot{u}_p\}$ at t = 0. Nonzero initial conditions are only permitted for a direct problem formulation. The specification of nonzero initial conditions for a modal formulation is not permitted because the selection of a number of modes less than the total number of degrees of freedom places constraints on the physical points that are incompatible with an arbitrary set of displacement values.

In a direct formulation, the initial conditions for all unspecified degrees of freedom are assumed to be zero, so that the user must be careful to provide a complete set of initial conditions. Initial conditions specified for points that are eliminated by constraints or by the Guyan reduction are ignored.

The initial conditions presented to the integration algorithm, Equations 32 and 33, are the values of $\{u_0\}$, $\{u_{-1}\}$, $\{P_0\}$ and $\{P_{-1}\}$. Two alternative starting methods have been provided, each having its own advantages. The first method is as follows:

The values of $\{u_{-1}\}$, and $\{P_{-1}\}$ are computed on the assumption that the acceleration for t < 0 is zero. Since the mass matrix [M] may be (and often is) singular, the evaluation of the acceleration at t = 0, from the load at t = 0, may be impossible in any case.

793

$$\{u_{-1}\} = \{u_o\} - \{\dot{u}_o\}\Delta t \quad , \tag{46}$$

$$\{P_{-1}\} = [K]\{u_{-1}\} + [B]\{\dot{u}_o\} \ . \tag{47}$$

In addition, the load specified by the user at $t = 0$ is replaced by

$$\{P_o\} = [K]\{u_o\} + [B]\{\dot{u}_o\} \quad , \tag{48}$$

which, in effect, sets the acceleration to zero at $t = 0$. The main reason for doing so is to avoid the "ringing" of massless degrees of freedom that are subjected to step loads. Consider, for example, that $[M]$, $[B]$, and $[N]$ are zero in Equation 31, which then becomes

$$\frac{1}{3}[K]\{u_{n+2} + u_{n+1} + u_n\} = \frac{1}{3}\{P_{n+2} + P_{n+1} + P_n\} \ . \tag{49}$$

If the values for $\{P_{-1}\}$ and $\{P_o\}$ given by Equations 47 and 48 are substituted into Equation 49, then

$$\{u_1\} = [K]^{-1}\{P_1\} \quad , \tag{50}$$

i.e., $\{u_1\}$ is the correct static solution for any initial conditions. Solutions at subsequent time steps will also be correct. On the other hand, suppose that $P_o = P_1 = P_2 \ . \ . \ . = 1$ and that the initial conditions are $u_o = \dot{u}_o = 0$. Then $u_{-1} = 0$, from Equation 46, and the sequence of values of u_n, considering only a single degree of freedom, are

$$\left.\begin{aligned} u_o &= 0 \ , \\ u_1 &= \frac{2}{K} \ , \\ u_2 &= \frac{1}{K} \ , \\ u_3 &= 0 \ , \\ u_4 &= \frac{2}{K} \ , \\ u_5 &= \frac{1}{K} \ , \\ u_6 &= 0 \ , \\ &\text{etc.} \end{aligned}\right\} \tag{51}$$

Thus the solution continues to "ring" indefinitely. The example cited is quite practical because the application of a load to a point which is restrained by a very stiff spring is often used to simulate the time history of an enforced displacement.

Although the starting algorithm shown in Equations 46 to 48 eliminates the problem of massless degrees of freedom, an approximation error is intorduced in the initial velocities. Note that the initial load, P_0, given by the user is ignored and the initial acceleration is assumed to be zero. These assumptions are approximate for the general case in which an initial load is applied to mass-loaded degrees of freedom.

For example, consider a single degree-of-freedom system having only mass and initial conditions $u_0 = \dot{u}_0 = 0$, and a constant applied load $P(t) = M$. Using Equations 47 and 48 to determine P_0 and P_{-1}, the calculated and theoretical sequence of displacements and velocities are

	Calculated		Theoretical	
Step	Displacement	Velocity	Displacement	Velocity
0	0	1/6 Δt	0	0
1	.33 Δt^2	2/3 Δt	.5 Δt^2	Δt
2	1.33 Δt^2	1.5 Δt	2.0 Δt^2	$2\Delta t$
3	3.33 Δt^2	2.5 Δt	4.5 Δt^2	$3\Delta t$
4	6.33 Δt^2	3.5 Δt	8.0 Δt^2	$4\Delta t$
5	10.33 Δt^2	4.5 Δt	12.5 Δt^2	$5\Delta t$

Note that the calculated velocities lag the theoretical velocities by $\Delta t/2$. In effect, a constant velocity error of $\frac{P\Delta t}{2M}$ has been introduced by ignoring the initial load. Because of this "drift", the displacement errors become progressively larger. Hence, the user is cautioned to beware when the initial loads are significant. If the structure is initially at rest, better answers are obtained using the second starting method option, described below.

To correct the above errors, which occur when the initial loads are dominant, a second (or alternate) starting method is provided. In this method, the initial load at $t = 0$ is assumed to be discontinuous, i.e., at $t < 0$ the load is described by Equation 48, at $t \geq 0$ the user input load is included. Thus

$$P_0 = \frac{1}{2}\left\{P_{ou} + [K]\{u_0\} + [B]\{\dot{u}_0\}\right\} \quad , \tag{52}$$

where P_{ou} is the initial user-defined load and $\{u_0\}$, $\{u_{-1}\}$, and $\{P_{-1}\}$ are the same as in the first method. The resulting equation for the first time step, obtained by substitution in Equation 31, is

$$\left[\frac{1}{\Delta t^2} M + \frac{1}{2\Delta t} B + \frac{1}{3} K\right] \{u_1\} =$$

$$\frac{1}{6}\{P_{ou} + 2P_1\} - \frac{1}{6}[K]\{u_0\} + \frac{1}{2\Delta t}[B]\{u_0\} + \frac{1}{\Delta t^2}[M]\{u_0 + \dot{u}_0 \Delta t\}. \tag{53}$$

For all points except massless degrees of freedom, Equation 53 provides correct first order results. For the massless points, the effects of non-equilibrium initial conditions provide errors equal to one-half the errors given in Equation 51.

It is frequently advantageous to change the size of the time step one or more times during a solution. For example, higher frequency components may be damped out quickly so that after a while errors in the higher frequency modes become unimportant, and an ecomony without loss of accuracy can be obtained by increasing the size of the time step. At the time of the change, the system matrices $[A_0]$, $[A_1]$, $[L_2]$ and $[U_2]$, Equations 32 to 35, are recalculated and initial conditions $\{u_0\}$, $\{u_{-1}\}$, $\{P_0\}$ and $\{P_{-1}\}$ are set internally to restart the integration algorithm. The criterion used for selecting the new initial conditions is that the displacement, velocity and acceleration are continuous at the time of transition.

Let N be the last step with the previous time step, Δt_1. The formulas used for calculating the previous velocity and acceleration at the transition are

$$\left.\begin{aligned} \{\dot{u}_0\} &= \frac{1}{\Delta t_1}\{u_N - u_{N-1}\} \quad , \\ \{\ddot{u}_0\} &= \frac{1}{\Delta t_1^2}\{u_N - 2u_{N-1} + u_{N-2}\} \quad . \end{aligned}\right\} \tag{54}$$

The initial conditions for the new integration are

$$\{u_0\} = \{u_N\} \quad , \tag{55}$$

$$\{u_{-1}\} = \{u_N\} - \Delta t_2 \{\dot{u}_0\} + \frac{1}{2} \Delta t_2^{\,2}\{\ddot{u}_0\} \quad , \tag{56}$$

$$\{P_0\} = \{P_N\} \quad , \tag{57}$$

$$\begin{aligned}\{P_{-1}\} &= [K]\{u_{-1}\} + [B]\{\dot{u}_{-1}\} + [M]\{\ddot{u}_{-1}\} \quad , \\ &= [K]\{u_{-1}\} + [B]\{\dot{u}_0 - \Delta t_2 \ddot{u}_0\} + [M]\{\ddot{u}_0\} \quad , \end{aligned} \tag{58}$$

where Δt_2 is the new time step. Note the assumption, in Equations 56 and 58, that the acceleration is uniform for $t \leq N\Delta t_1$. A smooth transition is thereby assured.

The outputs of the transient analysis module include velocities and accelerations as well as displacements. The formulas used for velocity and acceleration are

$$\{\dot{u}_n\} = \frac{1}{2\Delta t} \{u_{n+1} - u_{n-1}\} \quad , \tag{59}$$

$$\{\ddot{u}_n\} = \frac{1}{\Delta t^2} \{u_{n+1} - 2u_n + u_{n-1}\} \quad . \tag{60}$$

The output may be requested at even multiples of the integration time step, i.e., for every step, every second step, or every third step, etc. This feature affords some economy in output data preparation in cases where a small time step is needed for greater accuracy.

11.4.2 The CONTINUE Feature for Coupled Equations

In transient analysis, it is frequently necessary to continue the integration of the coupled equations beyond the last (or from any earlier intermediate) output time for which the solution was obtained in a previous run. Thus, the initial time for the new run is to be a specified output time of the previous run. Also, the displacements, velocities, and accelerations corresponding to the specified output time of the previous run are to be used as the initial conditions for the new run. A capability called the CONTINUE feature makes it possible to do this without re-executing the entire problem. It is available in both Rigid Formats 9 and 12 (Displacement Approach).

In order to use the CONTINUE feature, the user first requests a checkpoint of a coupled transient analysis problem in a normal manner. This run can terminate for any reason so long as the TRD module computes the solution for at least one output time and the UDVT (displacement-velocity-acceleration) file is successfully checkpointed. (The T0L (list of output times) file would have been previously checkpointed subsequent to the execution of the TRLG module). The TRD module also sets the parameter NC0L to be equal to the number of output time steps (which is also equal to one-third the number of columns in the UDVT matrix).

The user then restarts and activates the CONTINUE feature by changing any one or more of several cards either in the Case Control Deck (DL0AD, N0NLINEAR, TSTEP cards) and/or in the Bulk Data Deck (TSTEP, DAREA, DL0AD, F0RCE, etc.) that define either the dynamic loading and/or the time step selection.* This forces the re-execution of both the TRLG and TRD modules. Note that the dynamic loading and/or the time step selection** in the restart need not be the same as that used in the checkpoint run, but the structural model and the constraint data must be the same.

The CONTINUE feature causes the integration of the coupled equations to continue from the specified output time of the checkpoint run. The first displacement of the restart (or continued run) is given by

$$[D]\,\{u_1\} \;=\; \frac{1}{3}\,\{P_{-1} + P_0 + P_1\} + \{N_0\} + [C]\,\{u_n\} + [E]\,\{u_{-1}\} \quad , \tag{61}$$

where the matrices [D], [C] and [E] are as defined in Section 4.65.7.3 of the Programmer's Manual and,

$$\{P_0\} \;=\; [K]\,\{u_n\} + [B]\,\{\dot{u}_n\} + [M]\,\{\ddot{u}_n\} \quad , \tag{62}$$

$$\{u_{-1}\} \;=\; \{u_n\} - \Delta t\,\{\dot{u}_n\} + \frac{\Delta t^2}{2}\,\{\ddot{u}_n\} \quad , \tag{63}$$

$$\{\dot{u}_{-1}\} \;=\; \{\dot{u}_n\} - \{\ddot{u}_n\}\,\Delta t \quad , \tag{64}$$

and

$$\{P_{-1}\} \;=\; [M]\,\{\ddot{u}_n\} + [B]\,\{\dot{u}_{-1}\} + [K]\,\{u_{-1}\} \quad , \tag{65}$$

where [K], [B], and [M] are as defined in Section 11.4.1, $\{u_n\}$, $\{\dot{u}_n\}$, $\{\ddot{u}_n\}$ are the displacements, velocities and accelerations, respectively, at the specified output time t_n of the checkpoint run, and Δt is the initial time step for the restart. $\{P_1\}$ is the load at time $t = t_n + \Delta t$ and $\{N_0\}$ is the initial non-linear load.

---

*Additionally, if the user wishes to restart the integration from an intermediate (rather than the last) output time of the checkpoint run, the parameter NC0L should be appropriately reset (to correspond to the desired intermediate output time) by means of a PARAM statement just before the TRLG module in the DMAP sequence. This is done by means of a DMAP alter in the Executive Control Deck of the restart.

**The errors that are inherently introduced by the restart (Equations 63 through 65) may be minimized by selecting the initial time step in the restart to be the same as the time step used in the checkpoint run just before the restart.

The output of the restart does not include the solutions of the checkpoint run, but only those solutions that are computed by the restart. Also, any initial conditions specified in the data for the restart are ignored since the solution is continued by using the displacements, velocities, and accelerations corresponding to the specified output time of the checkpoint run as initial conditions.

INTEGRATION OF COUPLED EQUATIONS

REFERENCES

1. Newmark, N. M., "A Method of Computation for Structural Dynamics," Proc. ASCE, J. Eng. Mech. Div., EM-3, July 1959, pp. 67-94.
2. MacNeal, R. H., Electric Circuit Analogies for Elastic Structures, John Wiley and Sons, New York, 1962, p. 182.

800

11.5 INTEGRATION OF UNCOUPLED LINEAR EQUATIONS

When there are no direct input terms, i.e., when $[M_{dd}^2] = [B_{dd}^2] = [K_{dd}^2] = 0$, and when there are no nonlinear terms, the equations of motion using the modal formulation are uncoupled. Each modal coordinate, ξ_i, satisfies a separate second order differential equation,

$$m_i \ddot{\xi}_i + b_i \dot{\xi}_i + k_i \xi_i = P_i(t) \quad , \tag{1}$$

which is convenient to rewrite as

$$\ddot{\xi}_i + 2\beta \dot{\xi}_i + \omega_o^2 \xi_i = \frac{1}{m_i} P_i(t) \quad , \tag{2}$$

where

$$\beta = \frac{b_i}{2m_i} \quad , \tag{3}$$

$$\omega_o^2 = \frac{k_i}{m_i} \tag{4}$$

The general solution to Equation 2, expressed in terms of arbitrary initial conditions, $\xi_{i,n}$ and $\dot{\xi}_{i,n}$ at $t = t_n$, and a convolution integral of the applied load, is, for $t \geq t_n$,

$$\xi_i(t) = F(t-t_n)\, \xi_{i,n} + G(t-t_n)\, \dot{\xi}_{i,n} + \frac{1}{m_i} \int_{t_n}^{t} G(t-\tau)\, P_i(\tau)\, d\tau \quad . \tag{5}$$

The functions F and G are combinations of the homogeneous solutions

$$\xi_i(t) = e^{(-\beta \pm \sqrt{\beta^2 - \omega_o^2})(t-t_n)} \quad . \tag{6}$$

F and G satisfy, respectively, the initial conditions for unit displacement and unit velocity.

The applied load, $P_i(t)$, is calculated by the procedures described in Section 11.2 at times $t_n = t_o + nh$ where h is the time increment and t_o is the time of the last change in time increment. It is assumed that the load varies linearly between t_n and t_{n+1}, so that, in Equation 5,

PAGE 801 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

$$P_i(\tau) = P_{i,n} + \frac{\tau}{h}(P_{i,n+1} - P_{i,n}) \quad . \tag{7}$$

For this form of the applied load the integral in Equation 5 can be evaluated in closed form. The general form of the solutions at the next time step, $t = t_{n+1}$, in terms of the initial conditions at $t = t_n$ and the applied loads, is

$$\xi_{i,n+1} = F\,\xi_{i,n} + G\,\dot{\xi}_{i,n} + A\,P_{i,n} + B\,P_{i,n+1} \quad , \tag{8}$$

$$\dot{\xi}_{i,n+1} = F'\,\xi_{i,n} + G'\,\dot{\xi}_{i,n} + A'P_{i,n} + B'P_{i,n+1} \quad . \tag{9}$$

The coefficients are functions of the modal parameters, m_i, β, ω_o^2, and of the time increment, h. The uncoupled modal solutions are evaluated at all time steps by recurrent application of Equations 8 and 9. The accelerations, which may be requested as output, are calculated by solving for $\ddot{\xi}$ from Equation 2:

$$\ddot{\xi}_{i,n+1} = \frac{P_{i,n+1}}{m_i} - 2\beta\,\dot{\xi}_{i,n+1} - \omega_o^2\,\xi_{i,n+1} \quad . \tag{10}$$

The algebraic expressions for the coefficients in Equations 8 and 9 depend on whether the homogeneous solutions are underdamped ($\omega_o^2 > \beta^2$), critically damped ($\omega_o^2 = \beta^2$), or overdamped, ($\omega_o^2 < \beta^2$). In addition, a separate set of expressions is used for undamped rigid body modes, ($\omega_o = \beta_o = 0$). For reasons of numerical stability, the expressions for the critically damped case are used within a small interval near the critically damped condition, ($|\omega_o^2 - \beta^2| < \varepsilon_1\omega_o^2$), and the expressions for the undamped rigid body case are used within a small region near the rigid body condition, $(\omega^2 + \beta^2)^{1/2}\,h < \varepsilon_2$. The coefficients for all four cases are listed in Table 1.

Once the coefficients have been evaluated, the integration algorithm, Equations 8 and 9, proceeds very rapidly, provided that the time step, h, is not changed frequently. The algorithm is about as efficient as the finite difference integration algorithm, Equation 31 of Section 11.4, but is much more accurate. It is also more efficient, by a very large factor, than algorithms that employ convolution integrals over the entire preceding time history.

Table 1. Formulas for the Coefficients in Equations 8 and 9.

Define:

$$\omega^2 = |\omega_o{}^2 - \beta^2|$$

$$k = \omega_o{}^2 m_i$$

1. Underdamped Case, $(\omega_o{}^2 - \beta^2)/\omega_o{}^2 \geq \varepsilon_1 = 10^{-8}$

$$F = e^{-\beta h}(\cos\omega h + \frac{\beta}{\omega}\sin\omega h)$$

$$G = \frac{1}{\omega}e^{-\beta h}\sin\omega h$$

$$A = \frac{1}{hk\omega}\left\{e^{-\beta h}\left[\left(\frac{\omega^2 - \beta^2}{\omega_o{}^2} - h\beta\right)\sin\omega h - \left(\frac{2\omega\beta}{\omega_o{}^2} + h\omega\right)\cos\omega h\right] + \frac{2\beta\omega}{\omega_o{}^2}\right\}$$

$$B = \frac{1}{hk\omega}\left\{e^{-\beta h}\left[-\left(\frac{\omega^2 - \beta^2}{\omega_o{}^2}\right)\sin\omega h + \frac{2\omega\beta}{\omega_o{}^2}\cos\omega h\right] + \omega h - \frac{2\beta\omega}{\omega_o{}^2}\right\}$$

$$F' = -\frac{\omega_o{}^2}{\omega}\ e^{-\beta h}\sin\omega h$$

$$G' = e^{-\beta h}(\cos\omega h - \frac{\beta}{\omega}\sin\omega h)$$

$$A' = \frac{1}{hk\omega}[e^{-\beta h}\{(\beta + h\omega_o{}^2)\sin\omega h + \omega\cos\omega h\} - \omega]$$

$$B' = \frac{1}{hk\omega}[-e^{-\beta h}(\beta\sin\omega h + \omega\cos\omega h) + \omega]$$

2. Critically Damped Case, $\left|\dfrac{\omega_o{}^2 - \beta^2}{\omega_o{}^2}\right| < \varepsilon_1 = 10^{-8}$

$$F = e^{-\beta h}(1 + \beta h)$$

$$G = he^{-\beta h}$$

$$A = \frac{1}{hk}\left[\frac{2}{\beta} - \frac{1}{\beta}e^{-\beta h}(2 + 2h\beta + h^2\beta^2)\right]$$

$$B = \frac{1}{hk\beta}\left[-2 + \beta h + e^{-\beta h}(2 + \beta h)\right]$$

$$F' = -\beta^2 h e^{-\beta h}$$

$$G' = e^{-\beta h}(1 - \beta h)$$

$$A' = \frac{1}{hk}\left[e^{-\beta h}(1 + h\beta + h^2\beta^2) - 1\right]$$

$$B' = \frac{1}{hk}\left[1 - e^{-\beta h}(\beta h + 1)\right]$$

3. Overdamped Case, $(\omega_o^2 - \beta^2)/\omega_o^2 \leq -\varepsilon_1 = -10^{-8}$

$$F = e^{-\beta h}(\cosh\omega h + \frac{\beta}{\omega}\sinh\omega h)$$

$$G = \frac{1}{\omega}e^{-\omega h}\sinh\omega h$$

$$A = \frac{1}{hk\omega}\left\{e^{-\beta h}\left[\left(-\frac{\omega^2 + \beta^2}{\omega_o^2} - h\beta\right)\sinh\omega h - \left(\frac{2\omega\beta}{\omega_o^2} + h\omega\right)\cosh\omega h\right] + \frac{2\omega\beta}{\omega_o^2}\right\}$$

$$B = \frac{1}{hk\omega}\left\{e^{-\beta h}\left[\frac{\omega^2 + \beta^2}{\omega_o^2}\sinh\omega h + \frac{2\omega\beta}{\omega_o^2}\cosh\omega h\right] + \omega h - \frac{2\beta\omega}{\omega_o^2}\right\}$$

$$F' = -\frac{\omega_o^2}{\omega}e^{-\beta h}\sinh\omega h$$

$$G' = e^{-\beta h}(\cosh\omega h - \frac{\beta}{\omega}\sinh\omega h)$$

$$A' = \frac{1}{hk\omega}[e^{-\beta h}\{(\beta + h\omega_o^2)\sinh\omega h + \omega\cosh\omega h\} - \omega]$$

$$B' = \frac{1}{hk\omega}[-e^{-\beta h}(\beta\sinh\omega h + \omega\cosh\omega h) + \omega]$$

4. Undamped Rigid Body Modes, $(\omega^2 + \beta^2)^{1/2} h < \varepsilon_2 = 10^{-6}$

$$F = 1$$

$$G = h$$

$$A = \frac{h^2}{3m_i}$$

$$B = \frac{h^2}{6m_i}$$

$$F' = 0$$

$$G' = 1$$

$$A' = \frac{h}{2m_i}$$

$$B' = \frac{h}{2m_i}$$

11.6 TRANSIENT ANALYSIS BY THE FOURIER METHOD

Transient analysis by a Fourier transformation is separated into three phases. First, the loads (defined as a function of time) are transformed into the frequency domain. Second, the responses are computed in the frequency domain. Third, the responses (in the frequency domain) are transformed back to the time domain.

Two forms of the transform are considered, the Fourier series and the Fourier integral, which are defined in Table 1. Both methods must be numerically modified, which produces approximations. The inverse transform includes an infinite sum, for which only a finite number of terms are numerically evaluated. This approximation leads to truncation errors. The inverse Fourier integral must be numerically integrated, which leads to integration errors. The number of frequencies at which the integrand is evaluated is limited by cost of calculations.

Table 1. Definitions of Fourier Transforms

A. The Fourier Series

The basic time interval is $0 < t < T$, with the function periodic. The circular frequencies are given by

$$\omega_n = 2\pi n \Delta f \quad , \tag{1}$$

$$\Delta f = 1/T \quad . \tag{2}$$

The load transformation for a load at point a is

$$\tilde{P}_a(\omega_n) = \int_0^T P_a(t) e^{-i\omega_n t} \, dt \quad . \tag{3}$$

The response at point j is given by

$$\tilde{u}_j(\omega_n) = H_{ja}(\omega_n) \tilde{P}_a(\omega_n) \quad . \tag{4}$$

The response in the time domain is

$$u_j(t) = (\Delta\omega/\pi) \left[(\tfrac{1}{2}) u(0) + \sum_{n=1}^{\infty} \mathrm{Re}(\tilde{u}(\omega_n) e^{i\omega_n t}) \right] \quad , \tag{5}$$

where Re means "real part of."

PAGE 800 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

Table 1 (Continued)

B. The Fourier Integral

This is the limit as $T \to \infty$, $\Delta f \to 0$, $2\pi n \Delta f \to \omega$, of the Fourier series. Here, ω is a continuous variable.

$$\tilde{P}_a(\omega) = \int_0^\infty P_a(t) e^{-i\omega t}\, dt \quad , \tag{6}$$

$$\tilde{u}_j(\omega) = H_{ja}(\omega) \tilde{P}_a(\omega) \quad , \tag{7}$$

$$u_j(t) = (1/\pi) \int_0^\infty \mathrm{Re}(\tilde{u}(\omega) e^{i\omega t}) d\omega \quad . \tag{8}$$

11.6.1 Transformation of Loads

The user specifies loads in the same manner as given in Section 11.2. The two general forms are the tabular (Equation 3) (piecewise linear) and the general purpose function (Equations 4 and 5). The transformation is given by Equation 6 in Table 1 above, in which it is assumed that the user defines a function which vanishes for $t > T$.

For piecewise linear tabular functions, a table of pairs (x_i, Y_i) $(i = 1, N)$ is prescribed, which defines N-1 time intervals. In addition, an X1 shift and an X2 scale factor are allowed. Thus, the time-dependent load at point j is given by

$$P_j(t) = A_j Y_T\left(\frac{t - \tau_j - X1}{X2}\right) \quad . \tag{9}$$

A_j and τ_j are an amplitude factor and a delay which may depend upon the point which is loaded. The transformed load is

$$\tilde{P}_j(\omega) = A_j e^{-i\omega\tau_j}\, X2 \cdot \sum_{i=1}^{N-1} (x_{i+1} - x_i)(L_i Y_i + R_i Y_{i+1})/2 \quad , \tag{10}$$

where

$$L_i = e^{-i\omega(X1+X2\, x_i)} E_2(-i\omega X2(x_{i+1} - x_i)) \quad , \tag{11}$$

$$R_i = e^{-i\omega(X1+X2\, x_{i+1})} E_2(+i\omega X2(x_{i+1} - x_i)) \quad , \tag{12}$$

and

$$E_2(z) = \begin{cases} \frac{2}{z^2}\,(e^z - 1 - z) & \text{for large } z \\ 1 + \frac{z}{3} + \frac{z^2}{(3)(4)} + \cdots & \text{for small } z \end{cases} \tag{13}$$

The general purpose function is defined by

$$P_j(\omega) = \begin{cases} A_j\,\tilde{t}^n e^{\alpha\tilde{t}} \cos(2\pi f\,\tilde{t} + \phi) & 0 < \tilde{t} < T_2 - T_1 \\ 0 & \text{Otherwise} \end{cases} \quad , \tag{14}$$

where

$$\tilde{t} = t - T_1 - \tau_j \quad . \tag{15}$$

The value of n is restricted to be an integer for transient analysis by the Fourier method. The transform is given by

$$\tilde{P}_j(\omega) = A_j\, e^{-i\omega\tau_j}\,(R_2 + R_1)(T_2 - T_1)^{n+1}/2(n+1) \quad . \tag{16}$$

In Equation 16

$$R_2 = e^y\, E_{n+1}(z) \quad , \tag{17}$$

$$y = i\phi + (\alpha + i2\pi f)(T_2 - T_1) - i\omega T_2 \quad , \tag{18}$$

$$z = -[\alpha + i2\pi f - i\omega](T_2 - T_1) \quad . \tag{19}$$

R_1 is the same as R_2 except the signs of ϕ and f are reversed. Also,

$$E_K(z) = \begin{cases} \frac{K!}{z^K}\left(e^z - \sum\limits_{k=0}^{k=1} \frac{z^k}{k!}\right) \\ \text{or} \\ 1 + \frac{z}{K+1} + \frac{z^2}{(K+1)(K+2)} + \frac{z^3}{(K+1)(K+2)(K+3)} + \cdots \end{cases} \quad . \tag{20}$$

The second form is used for small values of z.

These loads, which appear in the form required for frequency response, are transformed to the modal coordinates exactly as in the modal frequency response method.

One other source of loads for aeroelastic problems is a one-dimensional gust. The same time dependencies are allowed as defined above; however, the amplitude (A_j) and delays (τ_j) for the aerodynamic elements are computed from areas, dihedrals, and coordinates in the flow direction.

11.6.2 Calculation of Frequency Response

The responses are computed by the method for coupled equations (see Section 12.2), since in aeroelasticity, which is the principal area of application, the aerodynamic matrices produce modal coupling. The aerodynamic matrices are computed at a few frequencies, and automatically interpolated to the others for computational efficiency. For small problems, a special core-held equation solver is used to increase the efficiency.

11.6.3 Inverse Transformation of the Response

The response is found from numerical approximation to Equation 8 or from the Fourier series result (Equation 5) which can be thought of as a special form of approximation to the integral (Equation 8). Three approximate methods have been coded to evaluate the inverse. In all cases, the quantity $\tilde{u}(\omega)$ is first calculated at a set of frequencies, ω_i, by the frequency response module. The ω_i's do not need to be equally spaced. For all methods $\tilde{u}(\omega)$ is set equal to zero outside the range of ω's computed. The methods are:

Method 0 - Approximate $\tilde{u}(\omega)e^{i\omega t}$ as a constant in each interval. This method reduces to the Fourier series approximation, Equation 5, for equal frequency intervals.

Method 1 - Fit $\tilde{u}(\omega)$ with a piecewise linear function, and do not approximate $e^{i\omega t}$.

Method 2 - Fit $\tilde{u}(\omega)$ with a cubic spline function, and do not approximate $e^{i\omega t}$.

Consider Method 2. Solving the "three-moment equations," the second derivatives $\tilde{u}''(\omega)$ can be found at each ω for which a frequency response has been computed. Then, in any interval $\omega_i < \omega < \omega_{i+1}$,

$$\tilde{u}(\omega) = [\tilde{u}(\omega_i)\cdot s + \tilde{u}(\omega_{i+1})\cdot r] - [(\omega_{i+1} - \omega_i)^2/6]\cdot[\tilde{u}''(\omega_i)(s-s^3) + \tilde{u}''(\omega_{i+1})(r-r^3)] \ , \quad (21)$$

where

$$\left.\begin{aligned} r &= (\omega-\omega_i)/(\omega_{i+1} - \omega_i) \\ s &= 1 - r \end{aligned}\right\} . \quad (22)$$

Integrate Equation 8, using Equation 21 for $\tilde{u}(\omega)$, and sum over the intervals. Then collect the terms for each ω_n, with the result,

$$u(t) = (1/\pi) \sum_{n=1}^{N} \mathrm{Re}\left\{[C_n(t)\tilde{u}(\omega_n) + D_n(t)\tilde{u}''(\omega_n)]e^{i\omega_n t}\right\} \quad , \tag{23}$$

$$C_n(t) = \frac{\omega_n - \omega_{n-1}}{2} E_2\left(-it(\omega_n - \omega_{n-1})\right) + \frac{\omega_{n+1} - \omega_n}{2} E_2\left(it(\omega_{n+1} - \omega_n)\right) \quad , \tag{24}$$

$$D_n(t) = -\frac{(\omega_n - \omega_{n-1})^3}{24} G\left(it(\omega_n - \omega_{n-1})\right) - \frac{(\omega_{n+1} - \omega_n)^3}{24} G\left(it(\omega_{n+1} - \omega_n)\right) \quad . \tag{25}$$

For the first term in Equation 23 (n=1), use only the second terms on the right side of Equations 24 and 25. For the last term in Equation 23 (n=N), use only the first terms on the right side of Equations 24 and 25. Also,

$$G(z) = 2E_2(z) - E_4(z) \quad , \tag{26}$$

where $E_K(z)$ is given by Equation 20.

The above form for the inverse transform has two advantages. Numerical problems for small values of $t\Delta\omega$ are efficiently evaluated by choosing the series form of Equation 20. Also, the other two methods are easily derived as subcases. If u" terms are removed from Equation 21, a piecewise linear fit occurs; thus, Method 1 results are given by deleting $\tilde{u}''$ from Equation 23, i.e.,

$$u(t) = (1/\pi) \sum_{n=1}^{N} \mathrm{Re}\left[C_n(t)\tilde{u}(\omega_n)e^{i\omega_n t}\right] \quad , \tag{27}$$

with C_n defined by Equation 24. Method 0 results if we repl[illegible]$_2$ by 1.0 in Equation 24.

The above procedure for Method 0 always multiplies the first and last term in the series by 1/2. In order to force agreement with the Fourier series, which is the limiting case of equal frequency intervals, the first term in the series is multiplied by 1/2 only if the value of the first frequency is zero.

Other special considerations are given the equal frequency interval case. If all Δf's are equal, and the first frequency is an integer multiple of Δf, the time step Δt is adjusted to make Δf·Δt = 1/integer, which reduces the number of distinct values of sin $\omega_n t$ and cos $\omega_n t$ used in Equation 27. Also, $C_n(t)$ and $D_n(t)$ (Equations 24 and 25) become independent of n and do not need to be computed at every frequency.

11.6.4 Practical Considerations for Use of the Fourier Methods

Some important practical considerations must be observed to use these methods successfully. To illustrate one problem, consider the response of a simple damped oscillator to a pulse (Figure 1). The upper three curves show the pulse and the response of the system if it is very stable and then slightly stable. Using the Fourier method, the pulse is replaced by a series of pulses, with period $1/\Delta f$. As can be seen, this gives good results if the system is very stable, but an incorrect impression if the system is slightly stable. Thus, in order for the results of the Fourier method to be valid:

a. The system should be stable,

b. The forcing functions should be zero for some time interval to allow decay, and

c. The frequency interval $\Delta f \leq 1/(T_{pulse} + T_{decay})$.

If the system has unstable modes, these will appear as a precursor before the pulse, just like a "stable mode in the reverse time." In general, Methods 1 and 2 are more accurate than Method 0. However, these methods introduce positive artificial damping into the result, which may lead to erroneous conclusions in stability studies. To see this, consider the function $C_n(t)$ in Equation 24. If equal frequency intervals are chosen, then

$$C_n(t) = \begin{cases} \Delta\omega \cdot \dfrac{1 - \cos(t\Delta\omega)}{\frac{1}{2}(t\Delta\omega)^2} = \Delta\omega\left(1 - \frac{1}{6}(t\Delta\omega)^2 + \ldots\right) & \text{Method 1} \\ \\ \Delta\omega & \text{Method 0} \end{cases} \quad . \qquad (28)$$

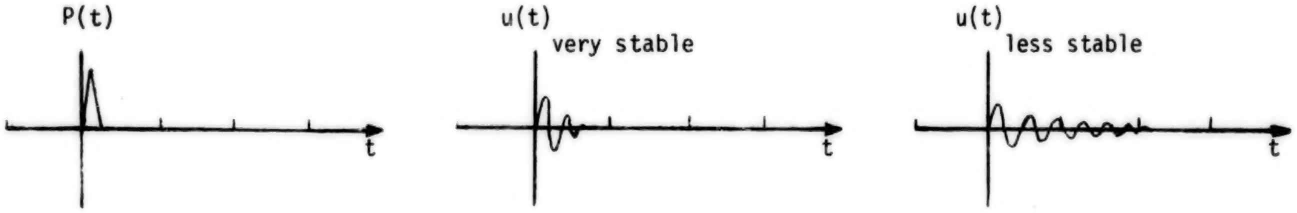
Thus, Method 1 (and also Method 2) produces a decaying envelope which the user may incorrectly interpret as additional damping.

The use of equal frequency intervals versus unequal intervals has been studied (Reference 1). It has been found that the combination of a few well-chosen values near the resonant frequencies and a uniformly spaced set of frequencies produces excellent results for lightly damped systems.

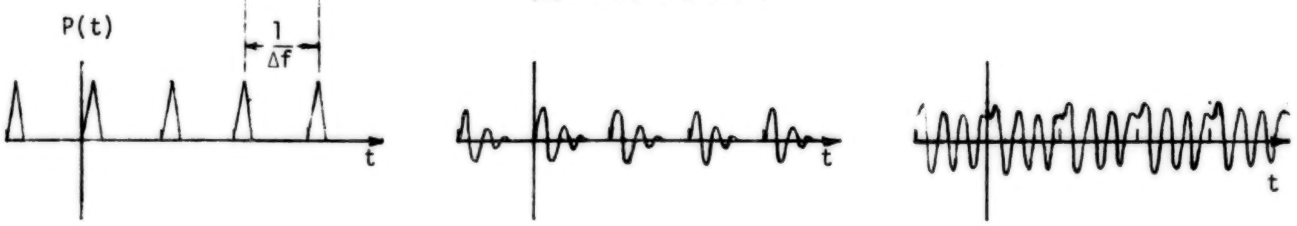
TRANSIENT ANALYSIS BY THE FOURIER METHOD

REFERENCE

1. Rodden, W. P., Harder, R. L., and Bellinger, E. D., "Aeroelastic Addition to NASTRAN." To be published as a NASA CR.


P(t)
t
u(t)
very stable
t
u(t)
less stable
t

(a) True Transient


P(t)
$\frac{1}{\Delta f}$
t
t
t

(b) Periodic Representation

Figure 1. Response of a Damped Oscillator to a Triangular Pulse.

12. FREQUENCY RESPONSE AND RANDOM ANALYSIS

12.1 INTRODUCTION

The ability to calculate the response of a system to steady sinusoidal excitation is important both because such excitation exists in the real world and because methods for solving other problems often depend on the knowledge of frequency response. The latter category of problems includes stability analysis (via Nyquist diagrams or Bode plots), random response analysis, and transient response analysis (via Fourier or Laplace transforms).

In NASTRAN, the calculation of frequency response and its use in random response analysis are performed by separate modules. There is at present no internal application of frequency response results to stability problems or to transient problems.

It is assumed that the reader is familiar with the theory of steady-state frequency response (which reduces linear differential equations to linear algebraic equations) and with the rudiments of random noise theory. The equations of motion are assumed to be linear and the statistical properties of the random excitation are assumed to be stationary with respect to time.

C-9

12.2 FREQUENCY RESPONSE

Figure 1 shows a simplified flow diagram for the frequency response module. The calculations are of two kinds: the generation of loads; and the solution of the dynamic equations to obtain a displacement response vector. Subsequent data reduction procedures are discussed in Sections 9.1 and 9.4.

In static analysis a number of automatic load generating subroutines (gravity load, pressure load, temperature-induced load, etc.) are employed in order to simplify the user's task of input data preparation. Similar schemes are not made available for dynamic analysis, mainly because of the great variety in the possible sources of dynamic loads. All that is done in frequency response analysis, and perhaps all that should be done, is to provide the user with a flexible input data format which allows for a varying degree of generality vs. simplicity.

The generation of dynamic loads can, at least in principle, be a formidable task. The most general form of loading in frequency-response analysis is one that varies arbitrarily in magnitude and phase with respect to the point of load application and with respect to frequency. Even for a problem of moderate size (say 200 degrees of freedom and 50 frequencies) the data set required to specify a general loading is large (200 x 50 = 10,000 entries). Thus it is desirable to have input data forms that accommodate special cases.

In NASTRAN, the dynamic load vector that is used in a specific subcase, $\{P_j^c\}$, is constructed as a combination of component load sets, $\{P_j^k\}$.

$$\{P_j^c(\omega)\} = \sum_k S_{ck}\{P_j^k(\omega)\} \quad , \tag{1}$$

where S_{ck} is a factor that gives the proportion of component load set k used in combined load set c. The advantage of this system is that it facilitates the examination of different combinations of loads from different sources. The form provided for specifying a component load set is

$$\{P_j^k(\omega)\} = \{A_{jk}\, e^{i(\theta_{jk} - \omega\tau_{jk})}\}\cdot G_k(\omega) \quad , \tag{2}$$

where A_{jk}, θ_{jk} and τ_{jk} are tabulated coefficients that may be different for each degree of freedom (j). $G_k(\omega)$ is a tabulated complex function of frequency (with two alternate forms) that is linearly interpolated to the frequencies, ω_i, at which solutions are requested. The form provided by

817 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

Equation 2 is most useful for loads due to traveling waves (in air, under water, or in the earth). A meaningful interpretation of the symbols in Equation 2 for such problems is that $G_k(\omega)$ is the pressure produced by the wave, that A_{jk} is the exposed area associated with the jth degree of freedom, and that τ_{jk} is the travel time required for the wave to reach the jth degree of freedom.

The coefficients A_{jk}, θ_{jk} and τ_{jk} are tabulated separately and one list of coefficients may be referenced by several load sets. Complete generality can be obtained with Equations 1 and 2 by identifying each load set with a specific frequency, i.e., by specifying

$$\begin{aligned} G_k(\omega) &= 1 \quad , \quad \omega = \omega_k \quad , \\ G_k(\omega) &= 0 \quad , \quad \omega \neq \omega_k \quad . \end{aligned} \tag{3}$$

The loads are referred by the user to the displacement set, u_p, that includes all physical points (structural points, u_g, and extra points, u_e). The reduction to final form consists of applying single and multipoint constraints, the Guyan reduction, and in the case of a modal formulation, the transformation to modal coordinates. Except for the modal transformation, the steps are identical to those for the reduction of static loads to final form described in Section 3.6.2. The loads on extra points are simply included in their proper sequence in each intermediate vector. The specific operations are

1. Partition P_p into P_m, loads on points eliminated by multipoint constraints, and $\overline{P}_n$.

$$\{P_p\} = \left\{\begin{array}{c} \overline{P}_n \\ \hdashline P_m \end{array}\right\} \quad . \tag{4}$$

2. Apply multipoint constraints.

$$\{P_m\} = \{\overline{P}_n\} + [G_m]^T\{P_m\} \tag{5}$$

3. Partition P_n into P_s, loads on points eliminated by single point constraints, and P_f.

$$\{P_n\} = \left\{\begin{array}{c} P_f \\ \hdashline P_s \end{array}\right\} \quad . \tag{6}$$

4. Partition P_f into P_o, loads on points eliminated by the Guyan reduction, and $\overline{P}_d$.

$$\{P_f\} = \begin{Bmatrix} \overline{P}_d \\ \hline P_o \end{Bmatrix} \quad . \tag{7}$$

5. Apply the Guyan reduction.

$$\{P_d\} = \{\overline{P}_d\} + [G_o]^T\{P_o\} \quad . \tag{8}$$

6. For a modal problem formulation only, transform P_d to the modal coordinates u_h, see Section 9.3.

$$\{P_h\} = [\phi_{dh}]^T\{P_d\} \quad . \tag{9}$$

The above operations can be time-consuming if there is a large number of load vectors to be transformed. If the vector of time constants, τ_{jk}, in Equation 2 is null, the relative magnitudes and phases of the loads at different grid points are independent of frequency. In that event the load transformation is applied once for each load set to the complex coefficients, $B_{jk} = A_{jk}\, e^{i\theta_{jk}}$, before the transformed load sets are combined. If τ_{jk} is not null for every load set in a particular combined load, $\{P_p{}^c\}$, the reduction procedure is applied to the combined load for each frequency.

For a direct formulation the equation to be solved is

$$[-\omega^2 M_{dd} + i\omega B_{dd} + K_{dd}]\{u_d\} = \{P_d\} \quad , \tag{10}$$

while for a general modal formulation, including direct (nonstructural) input the equation is

$$[-\omega^2 M_{hh} + i\omega B_{hh} + K_{hh}]\{u_h\} = \{P_h\} \quad . \tag{11}$$

The direct input matrices $[M_{dd}{}^2]$, $[B_{dd}{}^2]$, and $[K_{dd}{}^2]$ (see Section 9.3.3), are permitted to be dependent on frequency. If they are, an exit is made to the dynamic matrix assembler, GKAD, after each frequency (see Figure 1). The user can request solutions for a specified list of frequencies, or he can request a uniform or a logarithmic distribution of frequencies in a specified range. The selection of zero as a frequency permits the user to solve static problems with nonstructural, $[K_{dd}{}^2]$, terms.

The standard matrix decomposition and solution routines provided with the program are used in the solution of Equations 10 and 11. The following options are available: (a) double-precision arithmetic with row interchanges (partial pivoting); (b) double-precision without pivoting; (c) single-precision without pivoting. Option (a) is the default option.

If, in a modal formulation, there is no direct input, i.e., if $[M_{dd}{}^2] = [B_{dd}{}^2] = [K_{dd}{}^2] = 0$, the equations are uncoupled and a separate procedure is used as indicated in Figure 1. The solution for the modal coordinates is simply

$$\{\xi_i\} = \left\{\frac{P_i}{-m_i\omega^2 + ib_i\omega + k_i}\right\} \quad , \tag{12}$$

where m_i, b_i, and k_i are the generalized mass, damping and stiffness of the ith mode (see Section 9.3.4), and P_i is the ith component of $\{P_h\}$.

The evaluation of Equation 12 is trivial compared to the solution of Equation 10. It is not true, however, that a modal formulation of an uncoupled problem is always more efficient than a direct formulation, since it is first necessary to calculate the modes and to transform the loads and then, afterwards, to obtain physical coordinates from ξ_i by the modal transformation. The question of whether the modal approach is more efficient for any given problem depends on several factors, including the number of modes, the number of response frequencies, the bandwidth of the problem when formulated directly, and the presence or absence of nonstructural coupling.

The results obtained by the Frequency Response module are passed to the Dynamic Data Recovery module for further processing and thence to other modules in the data recovery chain, see Figure 1 of Section 9.1. The results that may be requested in either printed or plotted form include the components of displacement, acceleration and velocity; components of applied loads; and selected internal forces and stresses. The printed information may be sorted by frequency or by component. The plotted information consists of the magnitude and phase angle of any component plotted versus frequency. The magnitude scale is either linear or logarithmic and the frequency scale is either linear or logarithmic.

821

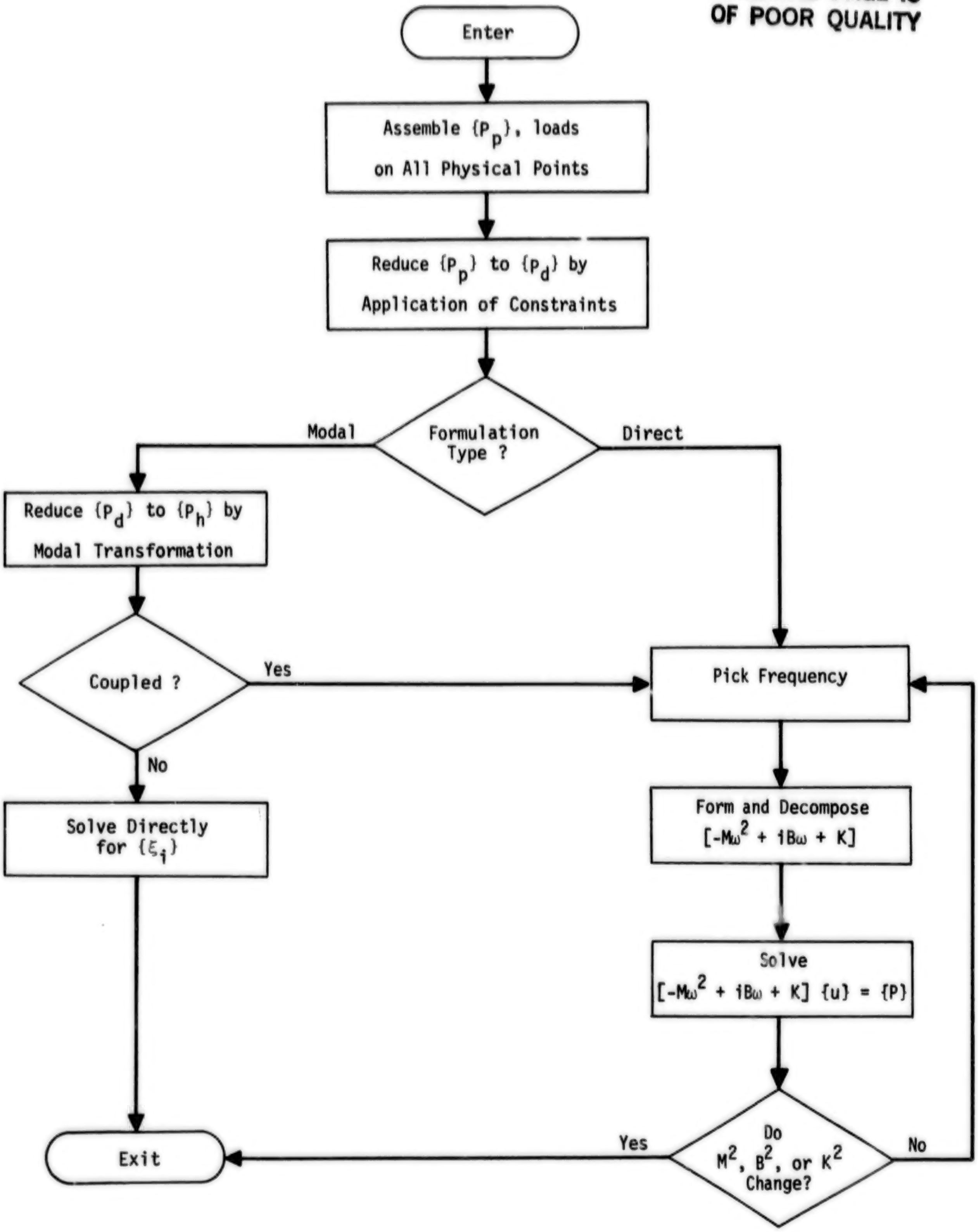

Enter
Assemble $\{P_p\}$, loads on All Physical Points
Reduce $\{P_p\}$ to $\{P_d\}$ by Application of Constraints
Formulation Type ?
Modal
Direct
Reduce $\{P_d\}$ to $\{P_h\}$ by Modal Transformation
Coupled ?
Yes
No
Solve Directly for $\{\xi_i\}$
Pick Frequency
Form and Decompose $[-M\omega^2 + iB\omega + K]$
Solve $[-M\omega^2 + iB\omega + K]\ \{u\} = \{P\}$
Do M^2, B^2, or K^2 Change?
Yes
No
Exit

Figure 1. Flow diagram for frequency response module, (FRD).

12.3 RANDOM ANALYSIS

The application of frequency response techniques to the analysis of random processes requires that the system be linear and that the excitation be stationary with respect to time. The theory includes a few important theorems which will be reviewed.

An important quantity in random analysis theory is the autocorrelation function, $R_j(\tau)$, of a physical variable, u_j, which is defined by

$$R_j(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_0^T u_j(t)\, u_j(t-\tau)dt \quad . \tag{1}$$

Note that $R_j(o)$ is the time average value of u_j^2, which is an important quantity in the analysis of structural failure. The power spectral density $S_j(\omega)$ of u_j is defined by

$$S_j(\omega) = \lim_{T \to \infty} \frac{2}{T} \left| \int_0^T e^{-i\omega t}\, u_j(t)dt \right|^2 \quad . \tag{2}$$

It may be shown (using the theory of Fourier integrals) that the autocorrelation function and the power spectral density are Fourier transforms of each other. Thus[1]

$$R_j(\tau) = \frac{1}{2\pi} \int_0^\infty S_j(\omega) \cos(\omega\tau)\, d\omega \quad , \tag{3}$$

from which follows the mean-square theorem,

$$\overline{u}_j^{\,2} = R_j(o) = \frac{1}{2\pi} \int_0^\infty S_j(\omega)d\omega \quad . \tag{4}$$

The expected value of the number of zero crossings with positive slope, or mean frequency, is a quantity of interest for fatigue analysis, airplane design for gusts, etc. This frequency, N_0, can be found from the power spectral density.

---

[1] The factor $\frac{1}{2\pi}$ in Equation 3 is omitted by some authors, and is sometimes replaced by other factors. The value of the factor depends on the definition of $S_j(\omega)$, Equation 2.

PAGE 823 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

$$N_0^2 = \frac{\left(\int_0^\infty (\omega/2\pi)^2 \, S_j(\omega) \, d\omega \right)}{\left(\int_0^\infty S_j(\omega) d\omega \right)} \quad . \tag{4a}$$

The frequency N_0 is thus the root mean square frequency, where the power spectral density is used as a weighting function.

The transfer function theorem (Reference 1) states that, if $H_{ja}(\omega)$ is the frequency response of any physical variable, u_j, due to an excitation source, Q_a, which may be a point force, a loading condition or some other form of excitation, i.e., if

$$u_j(\omega) = H_{ja}(\omega) \bullet Q_a(\omega) \quad , \tag{5}$$

where $u_j(\omega)$ and $Q_a(\omega)$ are the Fourier transforms of u_j and Q_a, then the power spectral density of the response, $S_j(\omega)$, is related to the power spectral density of the source, $S_a(\omega)$, by

$$S_j(\omega) = \left| H_{ja}(\omega) \right|^2 \bullet S_a(\omega) \quad . \tag{6}$$

Equation 6 is an important result because it allows the statistical properties (e.g., the autocorrelation function) of the response of a system to random excitation to be evaluated via the techniques of frequency response. Another useful result is that, if sources Q_1, Q_2, Q_3, etc., are statistically independent, i.e., if the cross-correlation function between any pair of sources

$$R_{ab}(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_0^T Q_a(t) \, Q_b(t-\tau) d\tau \quad , \tag{7}$$

is null, then the power spectral density of the total response is equal to the sum of the power spectral densities of the responses due to individual sources. Thus

$$S_j(\omega) = \sum_a S_{ja}(\omega) = \sum_a \left| H_{ja}(\omega) \right|^2 S_a(\omega) \quad . \tag{8}$$

If the sources are statistically correlated, the degree of correlation can be expressed by a cross-spectral density, S_{ab}, and the spectral density of the response may be evaluated from

$$S_j = \sum_a \sum_b H_{ja} H_{jb}^* S_{ab} \quad , \tag{9}$$

where H_{jb}^* is the complex conjugate of H_{jb}.

In applying the theory it is not necessary to consider the sources to be forces at individual points. An ensemble of applied forces that are completely correlated (i.e., a loading condition) may also be treated as a source. For example, a plane pressure wave from a specified direction may be treated as a source. Furthermore, the response may be any physical variable including internal forces and stresses as well as displacements, velocities and accelerations.

In NASTRAN, Random Analysis is treated as a data reduction procedure that is applied to the results of a Frequency Response analysis. The Frequency Response analysis is performed for loading conditions, $\{P_a\}$, at a sequence of frequencies ω_i. Normal data reduction procedures are applied to the output of the frequency response analysis module (see Figure 1 of Section 9.1), resulting in a set of output quantities, u_j. The calculation of power spectral densities and autocorrelation functions for the output quantites is performed in the Random Analysis module.

Figure 1 is a simplified flow diagram for the Random Analysis module. The inputs to the module are the frequency responses, $H_{ja}(\omega_i)$, of quantities u_j to loading conditions $\{P_a\}$ at frequencies, ω_i, and the auto- and cross-spectral densities of the loading conditions, S_a and S_{ab}. The response quantities, u_j, may be displacements, velocities, accelerations, internal forces, or stresses. The power spectral densities of the response quantities are calculated by Equation 9 or by Equation 6, depending on whether the loading conditions are correlated or uncorrelated. At user's option the spectral densities due to all sources, considered independent, may be combined by means of Equation 8.

The autocorrelation function is computed by the following approximation to Equation 3

$$R_j(\tau) = \frac{1}{2\pi\tau} \sum_{i=1}^{N-1} \left\{ \frac{S_j(\omega_{i+1}) - S_j(\omega_i)}{(\omega_{i+1} - \omega_i)\tau} \left[\cos(\omega_{i+1}\tau) - \cos(\omega_i\tau)\right] + S_j(\omega_{i+1}) \sin(\omega_{i+1}\tau) - S_j(\omega_i) \sin(\omega_i\tau) \right\} \quad , \tag{10}$$

which assumes that $S_j(\omega)$ varies linearly between ω_i and ω_{i+1}. The user specifies the sequence of values of τ. The rms value of the response, $\overline{u}_j$, is evaluated as the square root of a trapezoidal approximation to Equation 4, i.e.,

$$\overline{u}_j = \left[\frac{1}{4\pi}\sum_{i+1}^{N-1}[S_j(\omega_{i+1}) + S_j(\omega_i)](\omega_{i+1} - \omega_i)\right]^{1/2} \quad . \tag{11}$$

The rms frequency, N_o, is also evaluated from a trapezoidal curve for $S_j(\omega)$

$$N_o = \overline{r}_j/\overline{u}_j \quad , \tag{12}$$

where

$$\bar{r}_j = \left[\sum_{i=1}^{N-1}\Big(\alpha S_j(\omega_i) + \beta S_j(\omega_{i+1})\Big)(\omega_{i+1}-\omega_i)\right]^{\frac{1}{2}} \quad , \tag{13}$$

$$\alpha = (3\omega_i^2 + 2\omega_i\omega_{i+1} + \omega_{i+1}^2)/96\pi^3 \quad , \tag{14}$$

$$\beta = (\omega_i^2 + 2\omega_i\omega_{i+1} + 3\omega_{i+1}^2)/96\pi^3 \quad . \tag{15}$$

The power spectral densities, S_j, are plotted versus frequency and the autocorrelation functions, $R_j(\tau)$, are plotted versus the time delay, τ, at the user's request.

Cross-correlations and cross-spectral densities between different output quantities are not calculated.

The power spectral density function for atmospheric turbulence has been fit by several authors by analytic functions. Two of the commonly used functions are those due to Dryden and Von Karman (Reference 2). They can both be expressed by the equation,

$$S_a(\omega) = \frac{2W_g^2(L/U)[1+2(p+1)(kL\omega/U)^2]}{[1+(kL\omega/U)^2]^{p+3/2}} \quad , \tag{16}$$

where $S_a(\omega)$ = power spectral density (per Hz),
W_g = rms gust velocity,
ω = radian frequency,
L = scale of turbulence (length units),
U = airplane velocity (velocity units),
and parameter values are given by the following table:

	Dryden	Von Karman
k	1.0	1.339
p	1/2	1/3

A special data card is used to select this analytic form in NASTRAN. The user supplies W_g, L, and U and selects either the Dryden or the Von Karman parameters.

828

REFERENCES

1. Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., Aeroelasticity, Addison-Wesley Publishing Company, Reading, Mass., 1957, p. 823.

2. Taylor, J. (Ed.), Manual on Aircraft Loads, AGARDograph 83, Pergamon Press, 1965, pp. 200-202.

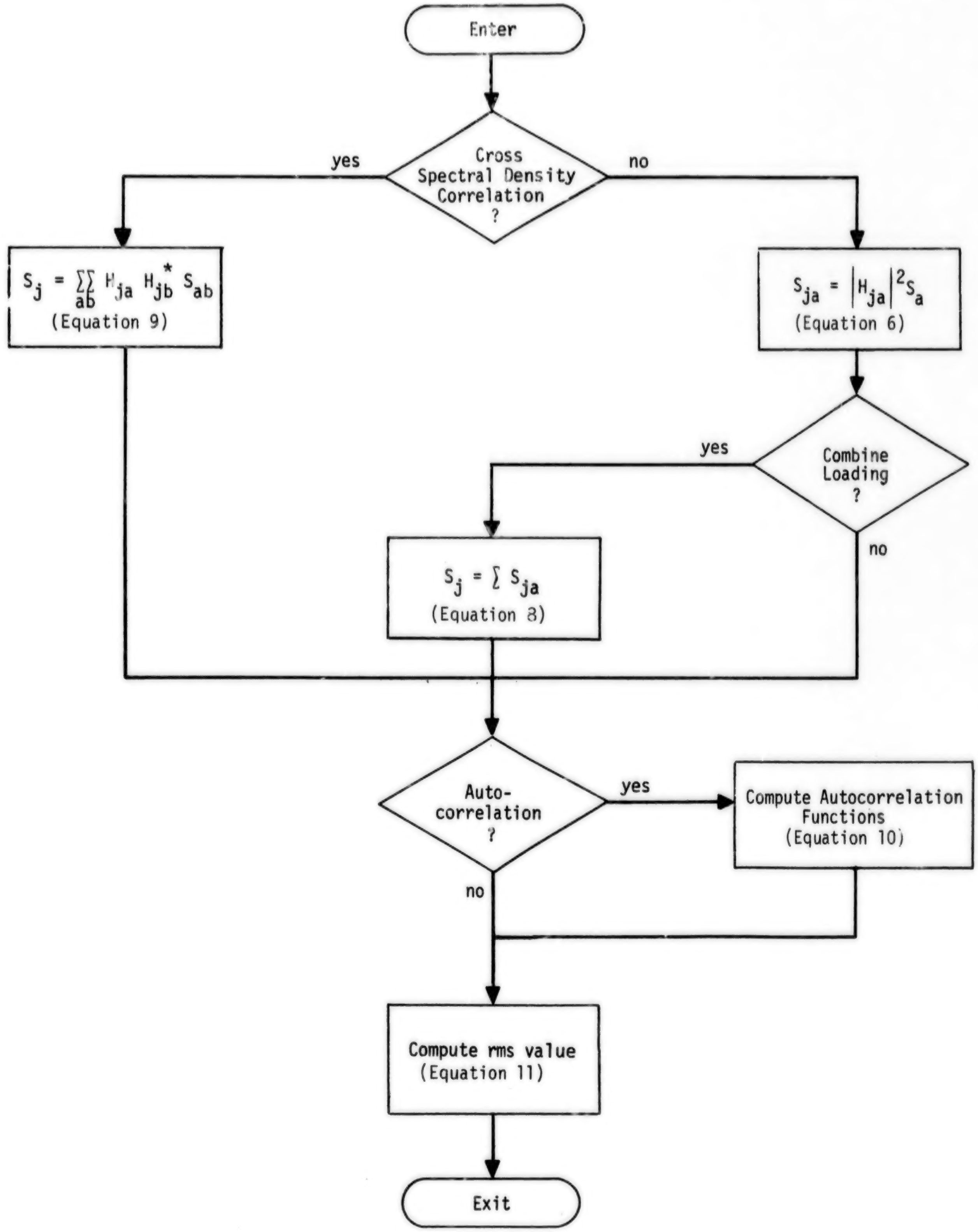

Enter
Cross Spectral Density Correlation ?
yes
no
$S_j = \sum_{ab}\sum H_{ja} H_{jb}^* S_{ab}$
(Equation 9)
$S_{ja} = |H_{ja}|^2 S_a$
(Equation 6)
Combine Loading ?
yes
no
$S_j = \sum S_{ja}$
(Equation 8)
Auto-correlation ?
yes
no
Compute Autocorrelation Functions
(Equation 10)
Compute rms value
(Equation 11)
Exit

Figure 1. Flow diagram for random analysis module.

13. COMPUTER GRAPHICS

13.1 STRUCTURE PLOTTER

Structures, as modeled by the finite element method employed in NASTRAN, readily yield to pictorial presentation by automatic plotters. The structural model is defined by a group of grid points, each of which has its location in space defined by a set of coordinates. These grid points are connected by structural elements with straight edges, which are easily drawn on the two-dimensional surface of the plotter output media.

For structures with many elements, plotting is a vitally important tool for detecting errors in the geometric location of grid points and in the connection of elements to grid points. Gross errors are prominently displayed, particularly if the structure has a regular geometry.

Structure plotting is also an important aid for visualizing the vibration and buckling mode shapes of geometrically complex structures, and the deflections of statically loaded structures. The ability to display the deflections at selected times of dynamically loaded structures is also available and may prove to be useful for some applications, in spite of the relatively large expense.

Since grid point coordinates are available in a common basic coordinate system (see Section 3.4), the operations of theoretical interest which the plot generating modules perform are:

1. to obtain the coordinates of a projection of the structural model on a user-selected two-dimensional surface by one of three available projection systems;
2. to convert the coordinates of points on the user-selected two-dimensional surface to plotter coordinates;
3. to scale the structural deformations in order to produce observable deflections of the structural model.

The theory used to produce orthographic, perspective, and stereoscopic projections will be described, followed by a discussion of model-to-plotter and deformation scaling.

A discussion of the means by which the NASTRAN user requests structure plots will be found in Chapter 4 of the User's Manual.

13.1.1 Structure Plotter Coordinate System and Orthographic Projection

In order to define the coordinates of an orthographic projection of the structural model, an R, S, T plotter coordinate system is defined as shown in Figure 1.

PAGE 831 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

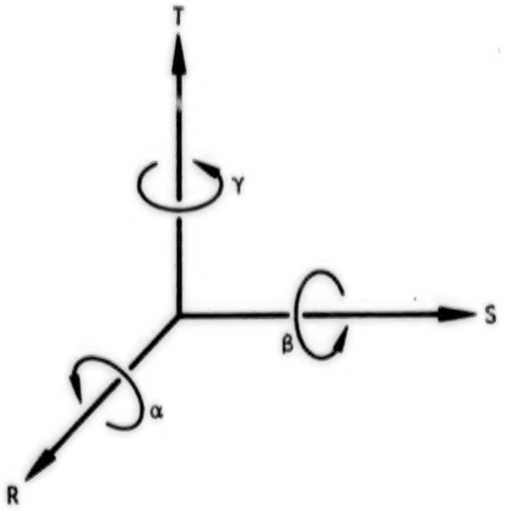


Figure 1. Plotter coordinate system.

The S-T plane is taken as the plane of projection. The structural model is defined in the basic coordinate system, which is denoted as the X, Y, Z coordinate system. The user specifies the position of the structural model with respect to the S-T projection plane by the angles γ, β and α. These angles position the X, Y, Z coordinate system with respect to the R, S, T coordinate system. The two coordinate systems are coincident for $\gamma = \beta = \alpha = 0$. The sequence in which the rotations are taken is crucial and has been arbitrarily chosen as γ, the rotation about the T-axis, followed by β, the rotation about the S-axis, followed by α, the rotation about the R-axis. Normally, α is not used since it does not affect the appearance of the S-T projection, but only its orientation on the page.

The orthographic projection is obtained by computing the S and T coordinates of each point having coordinates X, Y, Z from the transformation equation

$$\begin{Bmatrix} R \\ S \\ T \end{Bmatrix} = [A_\alpha][A_\beta][A_\gamma] \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} , \qquad (1)$$

833

where

$$[A_\gamma] = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} , \qquad (2)$$

$$[A_\beta] = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} , \qquad (3)$$

and

$$[A_\alpha] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} . \qquad (4)$$

In order to illustrate clearly the orientation process, Figure 2 shows a rectangular parallelepiped as it is rotated through the γ, β, α sequence. The final S-T plane shown in Figure 2d contains the desired orthographic projection.

13.1.2 Perspective Projection

In addition to the three angular relationships required for orthographic projection, the perspective projection requires knowledge of the vantage point in the R-S-T system (i.e., the three coordinates of the observer), and the location of the projection plane (plotter surface). The vantage point is selected by the user (or automatically by the program), and lies in the positive R half space as shown in Figure 3. The projection plane is chosen to lie between the observer and the S-T plane.

For each point, the coordinates S' and T' on the projection plane (see Figure 3) are obtained from the orthographic projection coordinates R, S, T by

$$\begin{Bmatrix} S' \\ T' \end{Bmatrix} = \begin{Bmatrix} S_o \\ T_o \end{Bmatrix} + \frac{d_o}{R_o - R} \begin{Bmatrix} S - S_o \\ T - T_o \end{Bmatrix} , \qquad (5)$$

where R_o, S_o, T_o are the coordinates of the vantage point and d_o is the separation distance between the vantage point and the projection plane.

13.1.3 Stereoscopic Projection

The stereoscopic effect is obtained through the differences in images received by the left and right eyes. Each is a perspective image, but with a different vantage point. The two vantage points are separated horizontally by 70 mm (2.756 inches), the nominal ocular separation standard used in commercially available stereoscopic cameras and viewers. Two plots are produced for viewing with a stereoscopic viewer.

13.1.4 Projection Plane-to-Plotter Transformations

Since the plotter surface is defined differently for each of the plotters used in NASTRAN, and since it is desirable to minimize the amount of special coding for each plotter, a common interface with the plotter routines is provided wherein the plotted surface is assumed to have a lower left corner defined by x, y coordinates (0,0) and an upper right corner defined by x, y coordinates (1,1). Plotter utility routines are utilized to convert from this system to each individual plotter.

Since the coordinates of points in any of the projection planes previously discussed may have arbitrary numerical values, a linear mapping of the form

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = [A_{ij}] \begin{Bmatrix} S \\ T \end{Bmatrix} + \begin{Bmatrix} x_o \\ y_o \end{Bmatrix} \tag{6}$$

is used, where the transformation constants $[A_{ij}]$ and x_o, y_o are determined as shown below so as to fill that portion of the x, y space required by the user or automatically selected by the program. In the case of a perspective projection, {S} and {T} are replaced by {S'} and {T'} . Since we desire no distortion of the plotted object and the orientation has already been specified, we set

$$[A_{ij}] = A \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . \tag{7}$$

Let

$$M = \frac{1}{2} \left[(T_{max} - T_{min}) - (S_{max} - S_{min}) \right] . \tag{8}$$

835

ORIGINAL PAGE IS
OF POOR QUALITY

Then, to fill the available x, y space we require, for $M > 0$,

$$\begin{aligned} 0 &= A\,T_{min} + y_o \\ 1 &= A\,T_{max} + y_o \\ m &= A\,S_{min} + x_o \\ 1-m &= A\,S_{max} + x_o \end{aligned} \tag{9}$$

or, for $M < 0$,

$$\begin{aligned} m &= A\,T_{min} + y_o \\ 1-m &= A\,T_{max} + y_o \\ 0 &= A\,S_{min} + x_o \\ 1 &= A\,S_{max} + x_o \end{aligned} \tag{10}$$

where m is the horizontal margin for $M > 0$ and the vertical margin for $M < 0$.

From (9) or (10) the quantities m, A, x_o, y_o, and thus the plotter coordinates, are determined.

13.1.5 Deformation Scaling

In plotting deformed structures, the components of displacement are added to the coordinates of the undeformed grid points to obtain the coordinates of the grid points of the deformed structure. Since the numerical magnitude of the physical structural displacements is usually much smaller than the size of the structure itself, additional scaling must be performed on the displacement vectors in order to obtain a viewable plot. In NASTRAN this is done by the user who specifies a value for the magnitude of the maximum structural deflection in units of length of the undeformed structural model. Thus, if max $\{T_{max} - T_{min}, S_{max} - S_{min}\}$ were, say, 1000 units, a specification of 50 would result in a plotted maximum deformation equal to 5% of the maximum plot size. In addition, the scale of the deformed structure, described in Section 13.1.4, is reduced by 5% to accommodate the deformation vectors.

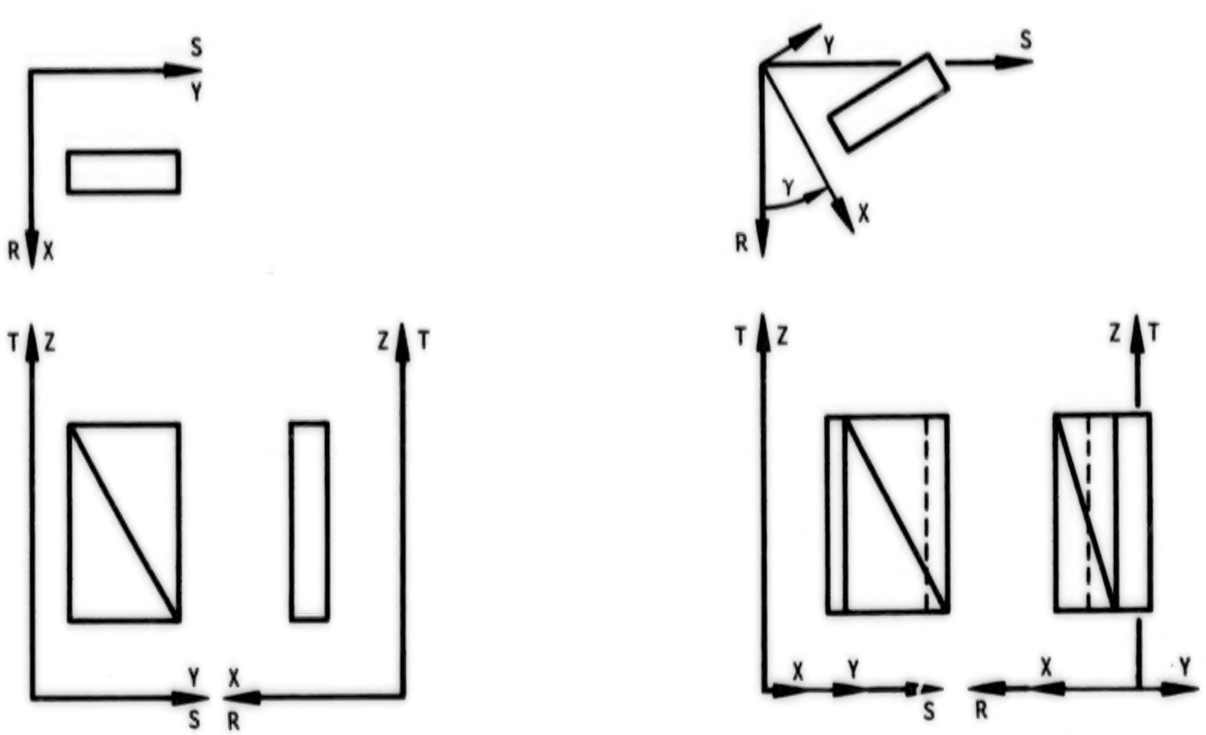
13.1.6 Structure Plotter Examples

Figures 4 and 5 show examples of plots generated by the NASTRAN Structure Plotter using the Stromberg-Carlson SC-4020 electronic plotter. Titling information usually presented on plots has been deleted from the figures.

Figure 4a shows a typical undeformed orthographic projection of a section of the structural model for a large dish antenna. This is about as detailed a model as one should attempt using the SC-4020 since the ability to "see" the object being depicted is marginal. Using one of the Calcomp plotters, however, a plot up to 30 inches square may be requested. Thus, considerably more detailed objects may be plotted without loss of clarity on the table plotters.

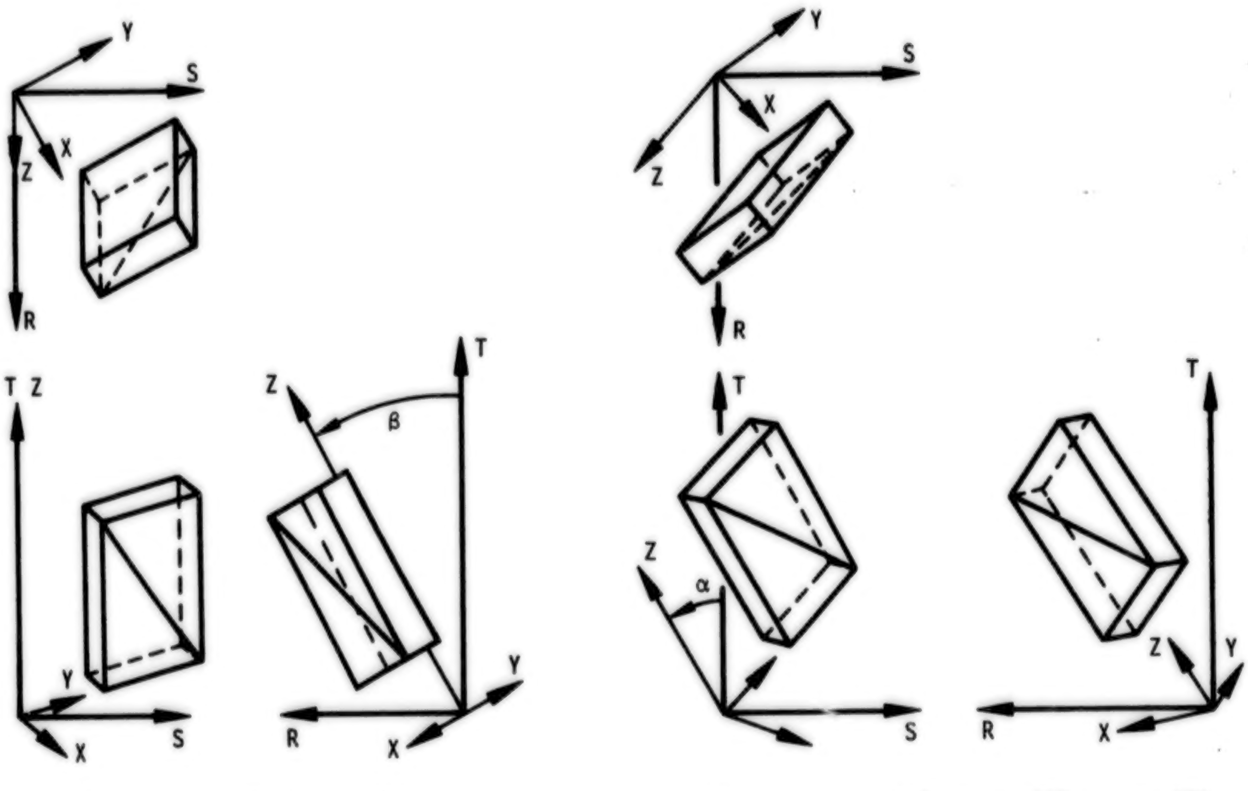
Figure 4b shows a perspective view of an undeformed 5 x 10 plate model. Grid point labels have been requested and are shown positioned adjacent to each grid point. A vector deformation request has resulted in the displacement patterns indicated. On the Calcomp plotters, a different pen could be used to draw the vectors with heavier or lighter lines than those of the undeformed shape, or with lines of a different color, or even on a transparent overlay. For the SC-4020 plotter, only a heavier line density can be employed to make the vector lines contrast with those of the undeformed structure.

Figure 5 shows two perspective views of plate models using the shape deformation option. These deformed shapes may be superimposed on the undeformed shape if desired.


S
Y
R
X
T
Z
Z
T
Y
S
X
R
Y
S
γ
R
X
T
Z
Z
T
X
Y
S
R
X
Y

(a) Initial Position

(b) γ = 30°


Y
S
Z
X
R
T Z
Z
β
T
Y
X
S
R
X
Y
Y
S
X
Z
R
T
Z
α
S
T
R
X
Z
Y

(c) γ = 30°, β = 30°

(d) = γ = 30°, β = 30°, α = 30°
(Plotter Will Plot on the S-T Plane)

Figure 2. Plotter — model orientation.

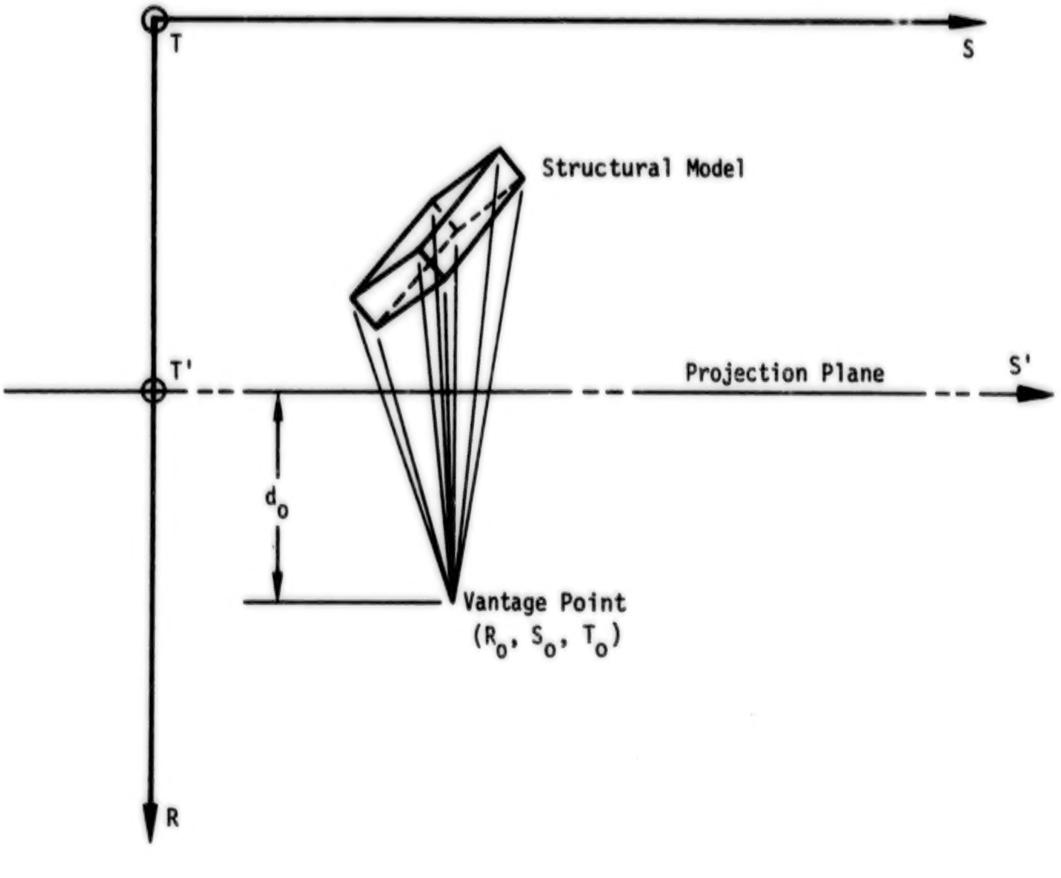
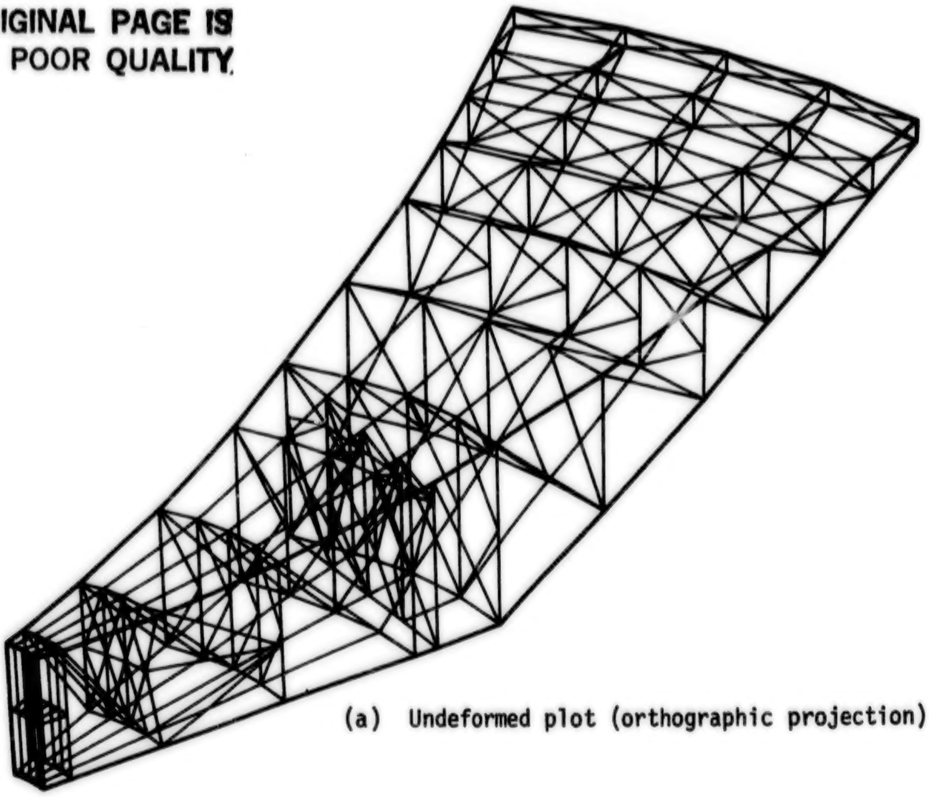
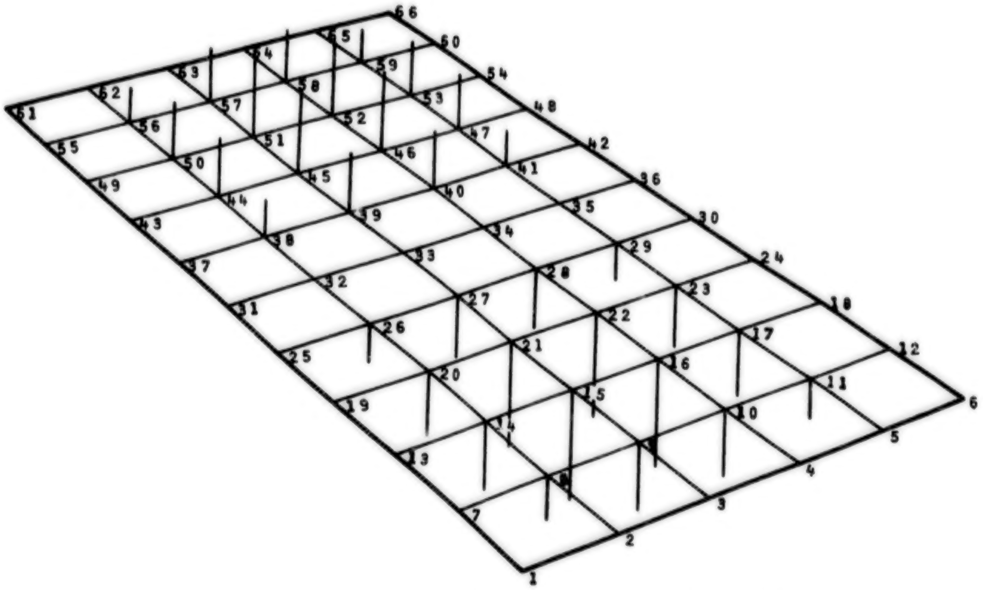

T
S
Structural Model
T'
Projection Plane
S'
d_0
Vantage Point
(R_0, S_0, T_0)
R

Figure 3. Perspective projection geometry.

839



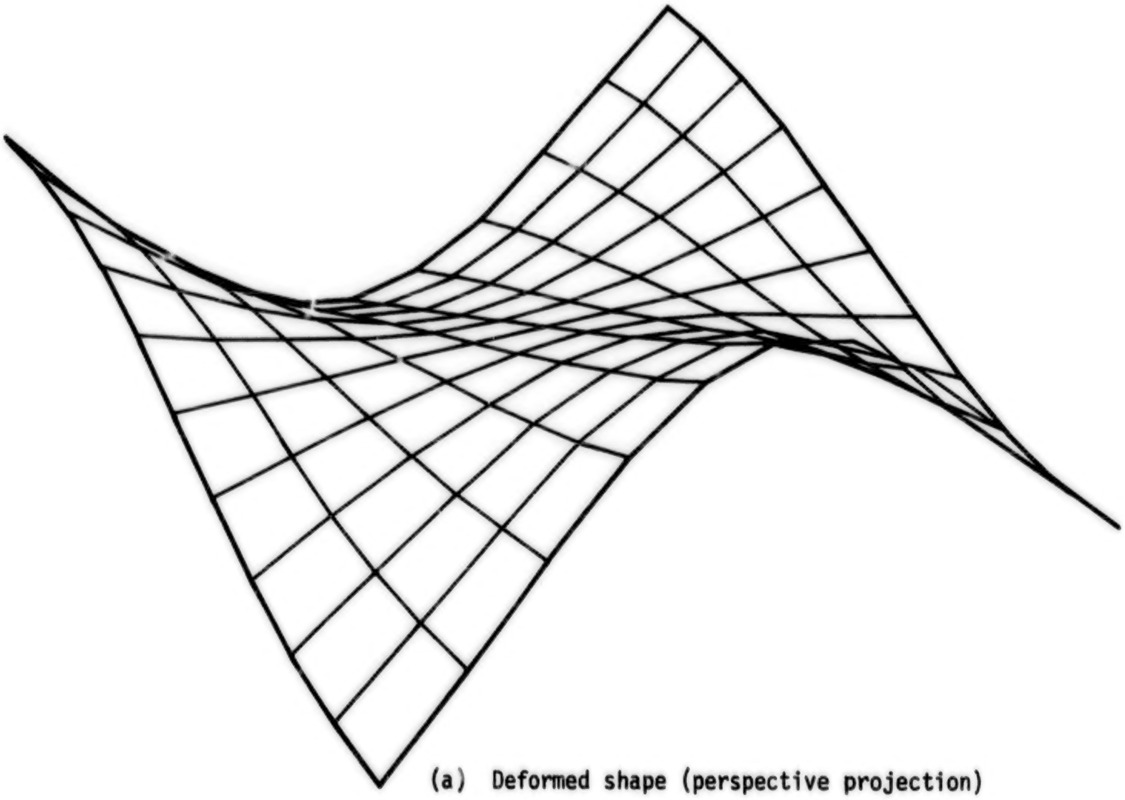
(a) Undeformed plot (orthographic projection)


61
62
63
64
65
66
55
56
57
58
59
60
49
50
51
52
53
54
43
44
45
46
47
48
37
38
39
40
41
42
31
32
33
34
35
36
25
26
27
28
29
30
19
20
21
22
23
24
13
14
15
16
17
18
7
8
9
10
11
12
1
2
3
4
5
6

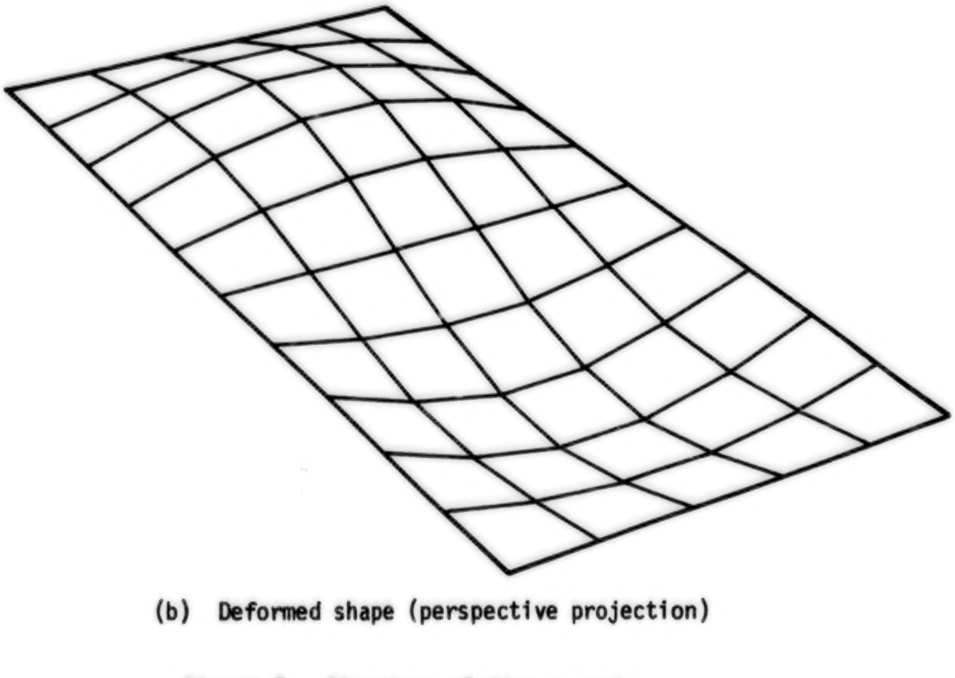
(b) Deformed plot using vectors (perspective projection)

Figure 4. Structure plotter examples.

ORIGINAL PAGE IS OF POOR QUALITY



(a) Deformed shape (perspective projection)



(b) Deformed shape (perspective projection)

Figure 5. Structure plotter examples.

13.2 CURVE PLOTTER

The NASTRAN curve plotter provides the user with the ability to generate x-y graphs of any available response quantities as functions of frequency (for frequency response analysis), time (for transient response analysis), or subcase (for static analysis). Scaling, axes, axis labeling, grid lines, logarithmic scales, and titling are all features automatically provided by the plotter. The user also may control those items if he wishes. Two basic forms are provided, the whole frame and the split frame. Whole frame curve plots utilize the whole sheet of graph paper or camera frame, whereas the split frame consists of an upper half frame and a lower half frame. The latter form is useful for plotting complex quantities resulting from frequency response analysis or for making comparisons.

On any of the frames described above, one or more curves may be plotted. If not controlled by the user, scaling will be accomplished so as to accomodate all curves on the same frame. Only one ordinate and one abscissa scale is provided for any single frame. The user has the option of drawing lines between the points of the tabular function, drawing symbols at the points of the tabular function or both. The use of different symbols for the several curves of a single frame is not provided, however.

The detailed description of the user input cards needed to request curve plots will be found in Section 4.3 of the User's Manual.

13.3 MATRIX TOPOLOGY PLOTTER

The NASTRAN Matrix Topology Plotter generates a picture showing the location of nonzero elements in a matrix. This feature is useful in displaying a stiffness matrix since the bandedness and location of active columns are important for efficiency as described in Section 2.2.

Use of the Matrix Topology Plotter is fully described in Section 5.3.2 of the User's Manual under Utility Module SEEMAT.

PRECEDING PAGE BLANK NOT FILMED

PAGE 843 INTENTIONALLY BLANK

14. SPECIAL MODELING TECHNIQUES

14.1 REPRESENTATION OF PART OF A STRUCTURE BY ITS VIBRATION MODES

It is sometimes required in a structural analysis to describe part of a structure by orthogonal vibration modes. In some instances structural information may not be available in other forms. The modal information may be derived either from other analyses or from vibration tests.

In the methods to be described, no special programming instructions and no special input data are used in describing the substructure to NASTRAN. The simulation of the part within the complete idealized model is accomplished by means of ordinary scalar elements and multipoint constraints. The particular arrangements of elements to be described are not the only possible ones, and the user is encouraged to employ variations of his own. The only general restrictions are those imposed by linearity, conservation of energy, and reciprocity.

The main purpose of this section is to illustrate the manner in which scalar elements and multipoint constraints can be used to simulate structural properties when they are expressed in abstract terms (e.g. by modal coordinates). A secondary purpose is to illustrate procedures that have been found useful in many practical situations.

When part of a structure is described by vibration modes, it must first be ascertained how the degrees of freedom at which it is connected to the remainder of the structure were supported when the vibration modes were measured (or computed). Three cases are distinguished:

1. All connection coordinates free.
2. All connection coordinates restrained.
3. Some connection coordinates free and some restrained.

The first condition is usually employed in vibration tests or analyses of large parts, such as an airplane fuselage. It is recognized that often it is not possible to achieve test conditions that are effectively free from restraint. (No such qualification applies of course to calculated modes). The second condition is usually employed in vibration tests or analyses of small parts, such as a horizontal stabilizer.

Case 1 will be discussed first, because it is the simplest. The required data are the vibration mode frequencies, ω_i, the mode shapes or eigenvectors, $\{\phi_i\}$, and the mass distribution of the part, expressed by the mass matrix $[M_p]$. The eigenvectors need not be normalized in any particular manner. Let the degrees of freedom at the points of connection to the remainder of the structure be designated by the vector $\{u_c\}$. Then the motions of these points are related to

PAGE 845 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

846

modal coordinates $\{\xi_i\}$ of the part by

$$\{u_c\} = [\phi_{ci}]\{\xi_i\} \tag{1}$$

(See Section 9.3.4 for a discussion of modal transformations.) The columns of $[\phi_{ci}]$ are the eigenvectors, $\{\phi_i\}$, abbreviated to include only the degrees of freedom at connection points, $\{u_c\}$. The usual approximation of including only a finite number of eigenvectors in $[\phi_{ci}]$ produces an idealized model for the part that is too stiff. Specification of the part is completed by calculating the generalized mass, m_i, stiffness, k_i, and damping, b_i, associated with each modal coordinate, ξ_i, as follows (See discussion in Section 9.3.4).

$$m_i = \{\phi_i\}^T[M_p]\{\phi_i\} \quad , \tag{2}$$

$$k_i = \omega_i^2 m_i \quad , \tag{3}$$

$$b_i = g_i m_i \omega_i \quad , \tag{4}$$

where g_i is a damping factor for the ith mode. Frequently g_i will not be accurately known.

The equation of motion for the generalized coordinate, ξ_i, is

$$(m_i p^2 + b_i p + k_i)\xi_i = \{\phi_{ci}\}^T\{f_c\} \quad . \tag{5}$$

where $\{f_c\}$ is the vector of forces applied to the substructure at the connection points, and $\{\phi_{ci}\}$ is the eigenvector $\{\phi_i\}$ abbreviated to include only the degrees of freedom at connection points.

Equations 1 through 5 contain all of the information required to describe the part. The way in which Equation 1 is used in the construction of the idealized model is to regard each of its rows as an equation of constraint between a constrained degree of freedom, u_c, and the free scalar points, $\{\xi_i\}$. m_i, k_i, and b_i are, respectively, concentrated scalar mass, stiffness, and damping elements connected to ξ_i. Figure 1 illustrates the interconnection of the elements in diagrammatic form.

The user prepares the idealized model of Figure 1 for NASTRAN by filling out data cards that declare the existence of scalar points, ξ_i; the values of scalar structural elements m_i, b_i, and k_i; and the coefficients in the equations of constraint. Since the information is input to the program at the element level, it is available for the solution of any static or dynamic problem by any of the rigid formats provided with NASTRAN.

ORIGINAL PAGE IS OF POOR QUALITY

REPRESENTATION OF PART OF A STRUCTURE BY ITS VIBRATION MODES

The derivation of an idealized model for the case when some or all of the connection points are restrained during measurement (or calculation) of the substructure modes is considerably more involved. A general solution, that was first devised in Reference 1, is developed below. The objective is to derive a set of relationships between the modal coordinates and the degrees of freedom at connection points (both free and restrained) that can be treated as equations of constraint. The modal mass, damping, and stiffness properties will, as in case 1, be simulated by scalar structural elements. The reader may find it helpful to review the notation and procedures described in Section 9.3 before proceeding.

Let the degrees of freedom of the substructure be partitioned into $\{u_a\}$, degrees of freedom that are free in the substructure modes, and $\{u_b\}$, degrees of freedom that are restrained in the substructure modes. The equations of motion for the substructure (without damping) can then be written as

$$\left\{\begin{array}{c} f_a \\ \hdashline f_b \end{array}\right\} = \left[\begin{array}{c:c} K_{aa} + M_{aa}p^2 & K_{ab} \\ \hdashline K_{ab}^T & K_{bb} \end{array}\right] \left\{\begin{array}{c} u_a \\ \hdashline u_b \end{array}\right\} \quad . \tag{6}$$

$\{f_a\}$ and $\{f_b\}$ are forces applied to the substructure. The mass of the substructure is assumed to be concentrated at the free coordinates, $\{u_a\}$, which include all coordinates not restrained in the substructure modes. Any substructure mass on the restrained coordinates, $\{u_b\}$, should be lumped into the remainder of the structure because the masses on restrained coordinates produce no effect during the vibration modes of the substructure. They are, therefore, ignored in the modal representation of the substructure. The stiffness matrix is partitioned in Equation 6 according to free and restrained coordinates. Note that $\{u_a\}$ contains the free connection coordinates as a subset.

The substructure mode shapes are described by a modal transformation between the free coordinates, $\{u_a\}$, and modal coordinates, $\{\xi_i\}$.

$$\{u_a\} = [\phi_{ai}]\{\xi_i\} \quad . \tag{7}$$

The corresponding generalized forces on the modal coordinates are

$$\{f_i\} = [\phi_{ai}]^T\{f_a\} \quad . \tag{8}$$

848

By virtue of the orthogonality property of vibration modes

$$[\phi_{ai}]^T[K_{aa} + p^2 M_{aa}][\phi_{ai}] = [k_i + m_i p^2] \quad , \tag{9}$$

where $[k_i]$ and $[m_i]$ are diagonal matrices of the modal coefficients computed by Equations 2 and 3. Thus, using Equations 7, 8, and 9 to transform Equation 6,

$$\left\{\begin{array}{c} f_i \\ \hline f_b \end{array}\right\} = \left[\begin{array}{c:c} k_i + m_i p^2 & \phi_{ai}{}^T K_{ab} \\ \hdashline K_{ab}{}^T \phi_{ai} & K_{bb} \end{array}\right] \left\{\begin{array}{c} \xi_i \\ \hline u_b \end{array}\right\} \quad . \tag{10}$$

It is convenient to separate the inertia forces from Equation 10, so that, defining

$$\{\overline{f}_i\} = \{f_i\} - [m_i]\{p^2 \xi_i\} \quad , \tag{11}$$

$$\left\{\begin{array}{c} \overline{f}_i \\ \hline f_b \end{array}\right\} = \left[\begin{array}{c:c} k_i & \phi_{ai}{}^T K_{ab} \\ \hdashline K_{ab}{}^T \phi_{ai} & K_{bb} \end{array}\right] \left\{\begin{array}{c} \xi_i \\ \hline u_b \end{array}\right\} \quad . \tag{12}$$

Equation 12 is a stiffness equation in standard form. A form that leads more directly to a useful physical model is obtained by placing ξ_i on the left-hand side. Thus

$$\left\{\begin{array}{c} \xi_i \\ \hline f_b \end{array}\right\} = \left[\begin{array}{c:c} k_i{}^{-1} & \psi_{ib} \\ \hdashline -\psi_{ib}{}^T & \overline{K}_{bb} \end{array}\right] \left\{\begin{array}{c} \overline{f}_i \\ \hline u_b \end{array}\right\} \quad , \tag{13}$$

where

$$[\psi_{ib}] = -[k_i]^{-1}[\phi_{ai}]^T[K_{ab}] \quad , \tag{14}$$

and

$$[\overline{K}_{bb}] = [K_{bb}] - [\psi_{ib}]^T[k_i][\psi_{ib}] \quad . \tag{15}$$

If the set of restrained points, $\{u_b\}$, is nonredundant, the matrix $[\overline{K}_{bb}]$, is null (see Section 5.7). This condition will be assumed. The matrix $[\psi_{ib}]$ is calculated from properties of the vibration modes as follows. During a vibration mode, $\{u_b\} = 0$, and the vector of forces acting on the constraints is, from Equation 13,

$$\{F_b\} = -\{f_b\} = [\psi_{ib}]^T\{\bar{f}_i\} = [\psi_{ib}]^T[k_i]\{\xi_i\} \tag{16}$$

Define $[K_{bi}]$ to be the matrix of forces on the constraints due to unit values of the modal coordinates while the substructure is vibrating in its normal modes.

$$\{F_b\} = [K_{bi}]\{\xi_i\} \quad . \tag{17}$$

Then, comparing Equations 16 and 17,

$$[\psi_{ib}] = [k_i]^{-1}[K_{bi}]^T \quad , \tag{18}$$

or, in other words, $[\psi_{ib}]$ is equal to $[K_{bi}]^T$ with each row divided by the appropriate element of $[k_i]$. $[\psi_{ib}]$ may also be used to define an auxiliary set of modal coordinates

$$\{\zeta_i\} = [\psi_{ib}]\{u_b\} \quad . \tag{19}$$

Then, from the top half of Equation 13,

$$\{\bar{f}_i\} = [k_i]\{\xi_i - \zeta_i\} \quad . \tag{20}$$

The free connection coordinates $\{u_c\}$ are a subset of $\{u_a\}$. The relation between $\{u_c\}$ and the modal coordinates $\{\xi_i\}$ is

$$\{u_c\} = [\phi_{ci}]\{\xi_i\} \quad , \tag{21}$$

where $[\phi_{ci}]$ is the appropriate partition of $[\phi_{ai}]$.

Equations 11, 19, 20 and 21 provide a complete description of the substructure. They are also used to construct the idealized model of the substructure, shown in Figure 2. The modal dampers, b_i, are placed across the modal springs, k_i, if they simulate structural damping. If they simulate damping due to the viscosity of a surrounding fluid environment, they should be placed between the modal coordinates and ground. The user may also, if he desires, retain some non-connection coordinates in the model in order to record motions at other points in the substructure. This is done by constructing constraints from additional rows of Equation 7. Equation 19 expresses a new set of constraint equations between the auxiliary modal coordinates and the degrees of freedom that are restrained in substructure modes.

The input data for the program consist of the coefficients of the equations of constraint, $[\phi_{ci}]$ and $[\psi_{ib}]$, the values of the scalar elements, m_i, b_i, and k_i, and their connections.

The techniques discussed above provide the capability for complete dynamic partitioning of a structure, since all of its parts, rather than a few, may be represented by their respective vibration modes (Reference 2). The general case shown in Figure 2 is particularly useful for this purpose. Consider, for example, the missile structure shown in Figure 3. The missile is physically partitioned with support conditions for the calculation of uncoupled vibration modes as shown in the figure. The first partition, (a), is unsupported while the others are cantilevered. The lumped element model for the composite system consists of parts with the form of Figure 2 connected in tandem. It is evident from the form of the lumped element model that the independent degrees of freedom consist of the modal coordinates $\{\xi_a\}$, $\{\xi_b\}$, $\{\xi_c\}$, etc. The displacement sets $\{u_{a,b}\}$, $\{\zeta_b\}$, $\{u_{b,c}\}$, etc., are all constrained. The dynamic equations, when written by the stiffness method, are banded with bandwidth equal to the number of modal coordinates in three successive partitions.

The user should be cautioned against an uncritical use of dynamic partitioning techniques. Use of a smaller number of modes as degrees of freedom to represent a dynamical system always results in a loss of mass, a loss of flexibility, or both. Procedures have been developed (References 1 and 3) for incorporating the "residual mass" or the "residual flexibility" into the analysis with substantial increase in accuracy. In general, however, established techniques for truncating the modes of a complete system do not automatically give good results when applied to partitions.

REPRESENTATION OF PART OF A STRUCTURE BY ITS VIBRATION MODES

REFERENCES

1. MacNeal, R. H., "Vibrations of Composite Systems," Office of Scientific Research, Report No. OSR TN 55-120, October 1954.

2. Hurty, W. C., "Dynamic Analysis of Structural Systems Using Component Modes," AIAA Journal, Vol. 3, No. 4, April 1965, pp. 678-685.

3. MacNeal, R. H., and Schwendler, R. G., "Optimum Structural Representation in Aeroelastic Analysis," ASD TR 61-680, January 1962.

ORIGINAL PAGE IS OF POOR QUALITY

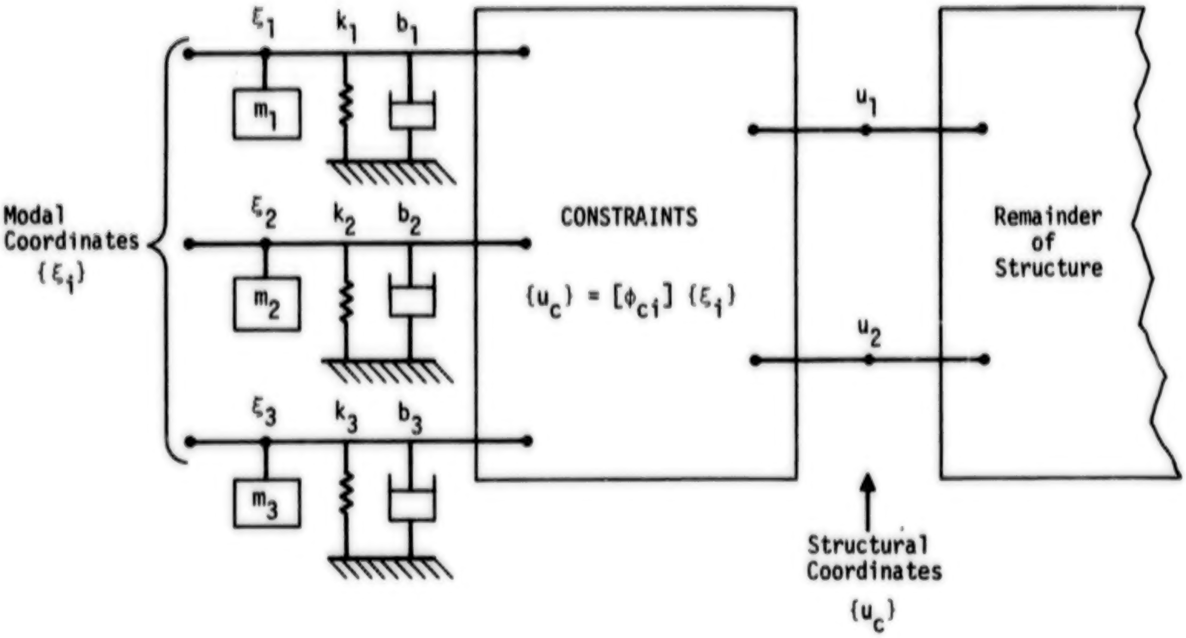

ξ_1
k_1
b_1
m_1
ξ_2
k_2
b_2
m_2
ξ_3
k_3
b_3
m_3
Modal Coordinates $\{\xi_i\}$
CONSTRAINTS
$\{u_c\} = [\phi_{ci}]\{\xi_i\}$
u_1
u_2
Remainder of Structure
Structural Coordinates $\{u_c\}$

Figure 1. Representation of part of a structure by its vibration modes when all connection points are free while the modes are calculated.

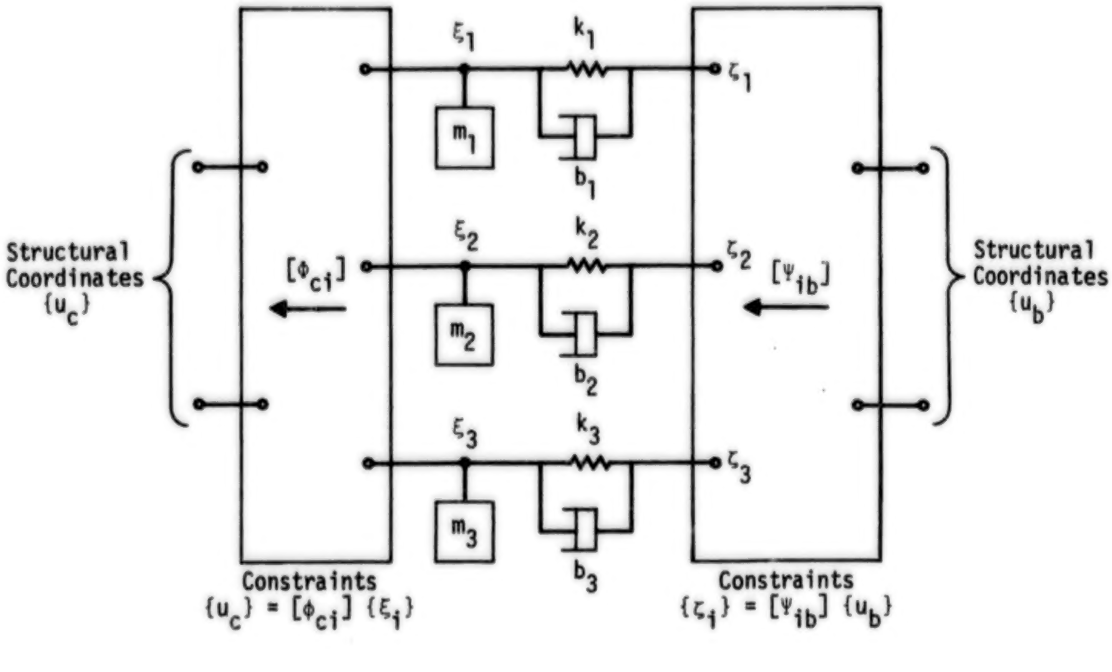
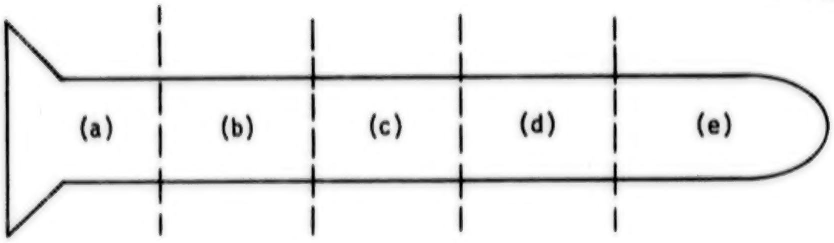
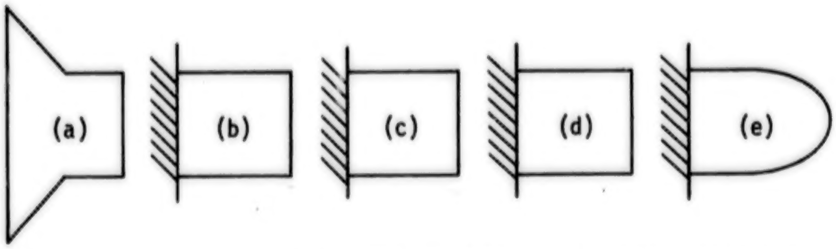

ξ_1
k_1
ζ_1
m_1
b_1
Structural Coordinates $\{u_c\}$
$[\phi_{ci}]$
ξ_2
k_2
ζ_2
$[\Psi_{ib}]$
Structural Coordinates $\{u_b\}$
m_2
b_2
ξ_3
k_3
ζ_3
m_3
b_3
Constraints $\{u_c\} = [\phi_{ci}]\{\xi_i\}$
Constraints $\{\zeta_i\} = [\Psi_{ib}]\{u_b\}$

Figure 2. Representation of part of a structure by its vibration modes in the general case when some connection points are free and some are rigidly constrained.

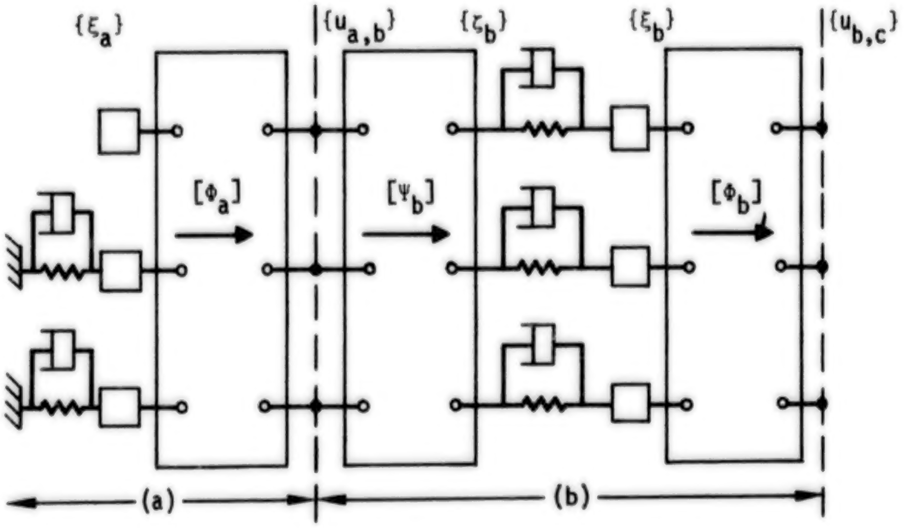
853



Missile structure, unpartitioned



Support conditions for partitions while calculating substructure vibration modes.


{ξ_a}
{u_a,b}
{ζ_b}
{ξ_b}
{u_b,c}
[Φ_a]
[Ψ_b]
[Φ_b]
(a)
(b)

Portion of composite model

Figure 3. Dynamic partitioning of missile structure.

14.2 REPRESENTATION OF CONTROL SYSTEMS

The principal means by which linear control systems are treated in NASTRAN are the Extra Points and the Transfer Function, described in Section 9.3.2. Nonlinear control systems employ, in addition, the Nonlinear Elements described in Section 11.2.

The general viewpoint is taken that the control system is an adjunct of the structure, rather than vice versa. Thus, the properties of the control system are expressed in the quadratic format $(Mp^2 + Bp + K)$ of dynamic structural analysis. Since some readers, including control system analysts, may not be familiar with the technique, an example that includes the major points is described below.

The variables that exist in control systems (voltages, valve positions, etc.) are assigned degrees of freedom, u_e, that are called extra points. The vector of extra points, $\{u_e\}$, is merged with $\{u_a\}$, the degrees of freedom at structural gridpoints, to form the dynamic analysis set $\{u_d\}$ (see Section 9.3.1). In the direct method of problem formulation, the extra points can be assigned sequence numbers such that they are interspersed with structural degrees of freedom. In the modal method of problem formulation they are collected together at the end of the augmented modal vector $\{u_h\}$ (see Section 9.3.4).

In order to facilitate the treatment of control systems, NASTRAN includes an input data format for the specification of transfer functions in the form

$$u_e = \frac{1}{b_0 + b_1 p + b_2 p^2} \sum_i (a_0{}^i + a_1{}^i p + a_2{}^i p^2)\, u_i \quad . \tag{9}$$

where u_e must be an extra point but u_i can be any degree of freedom. As explained in Section 9.3.2, Equation 9 is treated as a differential equation

$$(b_0 + b_1 p + b_2 p^2)\, u_e - \sum_i (a_0{}^i + a_1{}^i p + a_2{}^i p^2)\, u_i = 0 \quad , \tag{10}$$

which is incorporated into the general dynamic matrix equation

$$[K + Bp + Mp^2]\{u\} = \{P\} \quad . \tag{11}$$

The coefficients in Equation 10 thereby assume the roles of coefficients in the direct input stiffness, damping and mass matrices, $[K_{dd}{}^2]$, $[B_{dd}{}^2]$, and $[M_{dd}{}^2]$. The user may also add terms directly

PAGE 855 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

to these matrices by means of a separate direct matrix input format. It is also permitted to apply a load to an extra point, so that the vector {P} in Equation 11, includes the vector of such loads, $\{P_e\}$, as a subset.

As an example of the general approach, consider the control system shown in Figure 1 which might represent a simplified model of the autopilot of a launch vehicle. It contains many of the components found in control systems including attitude and rate sensors, signal conditioners, and a nonlinear mechanical actuator with local feedback. The structure is represented in Figure 1 by the displacement components which are sensed by the control system, or to which loads are applied.

The equations of the control system are listed in Table 1. The first six equations give the outputs of the devices labeled 1 to 6 in Figure 1 in terms of their inputs. Equation 7 defines the input signal, u_{10}, as a known function of time. Equation 8 states that the force on the structure, F_{11}, is a function of the input to the mechanical actuator, u_9.

In the analysis, an Extra Point is assigned to each of the new variables, u_4 - - - u_{10}. The coefficients of the transfer functions, expressed by Equations 1 to 8, are listed in Table 2. In the case of the input signal, Equation 7, a load equal to $\theta(t)$ is placed on coordinate 10, and the diagonal term in the stiffness matrix, b_0, is set equal to unity.

The mechanical force produced by the control system, F_{11}, is a nonlinear function of the input to the actuator, u_9, as shown in Figure 1. This function is most easily treated by the Arbitrary Function Generator element (see Section 11.2), for which the governing equation is

$$N_i(t) = S_i F(x_j) \quad , \qquad (12)$$

where $N_i(t)$ is the load applied to u_i,

S_i is an arbitrary (convenience) factor,

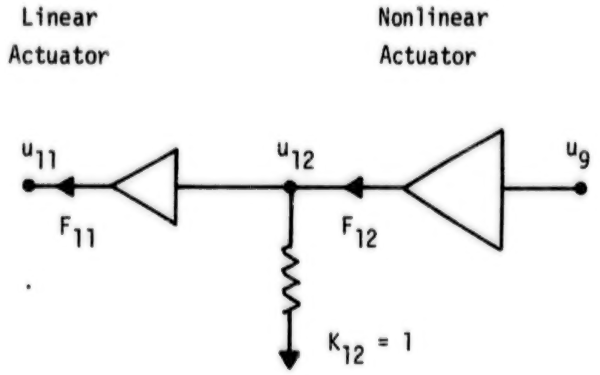
$F(x_j)$ is a tabulated function,

and x_j is any (permissible) displacement or velocity component.

If the actuator were linear, such that

$$F_{11} = K u_9 \quad , \qquad (13)$$

the effect could also be represented by placing -K in the element of the direct input stiffness matrix $[K_{dd}{}^2]$ corresponding to the row of u_{11} and the column of u_9. Nonlinear terms are not permitted in complex eigenvalue analysis or in frequency response analysis, so that the linear form of representation is required for such cases. Also, in a modal formulation of a transient response problem, it is not permitted to apply nonlinear loads directly to structural grid points. This restriction is imposed in order to avoid the modal transformation of nonlinear terms. If a modal formulation has been selected, the representation of the mechanical actuator in Figure 1 is modified as shown below.



An additional extra point, u_{12}, is created to which the nonlinear force $F_{12} = F(u_9)$ is applied. Two terms are added to the direct input stiffness matrix: the element of $[K_{dd}{}^2]$ corresponding to the row and column of u_{12} is set equal to +1; the element of $[K_{dd}{}^2]$ corresponding to the row of u_{11} and the column of u_{12} is set equal to -1. By this means the force on the structure, F_{11}, is made equal to F_{12}, and modal transformation of the nonlinear force is avoided.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 1. Equations of Example Control System

$$u_4 = \frac{1}{1 + \tau_1 p} (Apu_2 + Bpu_3) \quad , \tag{1}$$

$$u_5 = \frac{1}{1 + \tau_2 p} u_1 \quad , \tag{2}$$

$$u_6 = \frac{1}{1 + \tau_3 p} u_{10} \quad , \tag{3}$$

$$u_7 = \frac{A_o + A_1 p}{B_o + B_1 p + B_2 p^2} (C\, u_4 + D\, u_5 + E\, u_6) \quad , \tag{4}$$

$$u_8 = \frac{G\, u_{11}}{1 + \tau_4 p} \quad , \tag{5}$$

$$u_9 = u_7 - u_8 \quad , \tag{6}$$

$$u_{10} = \theta(t) \quad , \tag{7}$$

$$F_{11} = F(u_9) \quad . \tag{8}$$

TABLE 2. Transfer Function Table

Transfer Function No.	u_e	b_o	b_1	b_2	u_i	a_o	a_1	a_2
1	u_4	1	τ_1		u_2 u_3		A B	
2	u_5	1	τ_2		u_1	1		
3	u_6	1	τ_3		u_{10}	1		
4	u_7	B_o	B_1	B_2	u_4 u_5 u_6	$C\ A_o$ $D\ A_o$ $E\ A_o$	$C\ A_1$ $D\ A_1$ $E\ A_1$	
5	u_8	1	τ_4		u_{11}	G		
6	u_9	1			u_7 u_8	1 -1		
7	u_{10}	1						

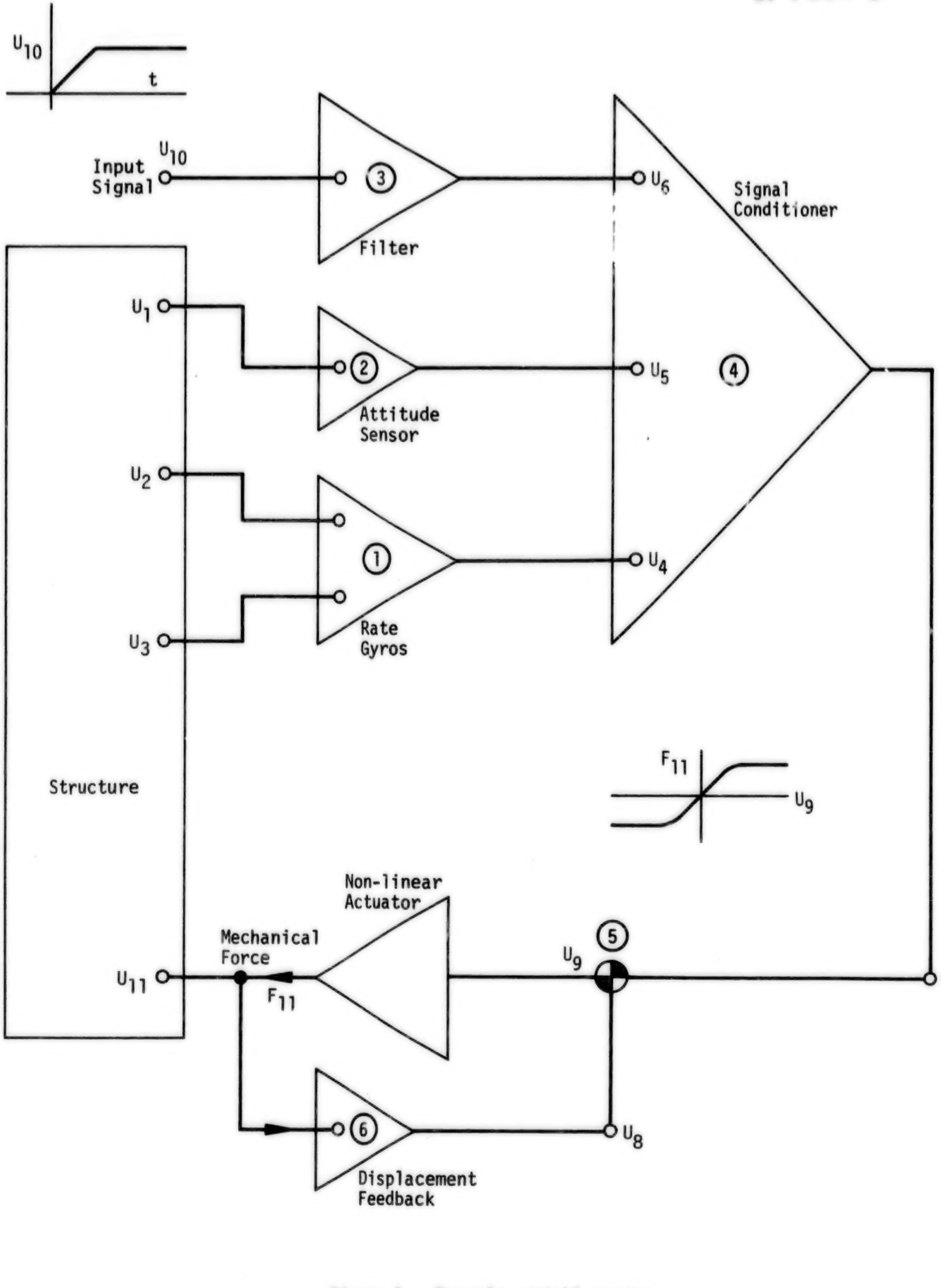

U_{10}
t
Input Signal
U_{10}
③
Filter
U_6
Signal Conditioner
Structure
U_1
②
Attitude Sensor
U_5
④
U_2
①
Rate Gyros
U_4
U_3
F_{11}
U_9
Non-linear Actuator
Mechanical Force
⑤
U_9
U_{11}
F_{11}
U_7
⑥
Displacement Feedback
U_8

Figure 1. Example control system.

801

15. ERROR ANALYSIS

15.1 SELECTION OF SIGNIFICANT FIGURES BASED ON ROUND-OFF ERRORS

One of the more important early decisions in the design of NASTRAN was the decision to use double-precision arithmetic (approximately sixteen decimal digits) in most of the calculations made with NASTRAN, including the formation of structural matrices and matrix decomposition. Since most existing structural analysis programs employ single-precision for stiffness matrix formation and matrix decomposition, an explanation of the reasons for the decision is in order. It should, perhaps, be emphasized that the decision was made on the assumption that a single-precision number has 27 binary bits (approximately eight decimal digits) rather than the 21 bits that are available with the IBM 360/370 computers, or the 48 bits that are available with the CDC 6000 series computers.

The only significant exceptions where single-precision is used in NASTRAN occur in calculations involving previously computed response vectors, such as in modal transformation and in the calculation of stresses from displacements.* Section 2 of the Programmer's Manual documents the precision of the matrix data blocks. If a matrix data block is in double-precision format, then it was generated using double-precision arithmetic. All table data blocks are in single-precision format.

It was known at the time the decision was made that NASTRAN would be called upon to solve structural problems that were large according to the then current standards. Previous experience with single-precision arithmetic was not, therefore, regarded as a reliable guide, and a brief analytical and experimental investigation was made of the effects of round-off errors in beam and framework problems. No attempt was made to develop a comprehensive view of the effects of grid point sequencing, stiffness tapering, different kinds of elements, etc., on the growth of roundoff error. It was known from previous experience that beams (particularly cantilever beams) have notoriously ill-conditioned stiffness matrices so that results based on the study of beams should be conservative for other structures. Since the decision to be made was whether or not to include eight additional decimal digits, only very rough accuracy was required. The results of the study are reported below.

Two different kinds of errors were considered: errors arising in the calculation of the elements of a stiffness matrix; and errors arising in the subsequent triangular decomposition of a stiffness matrix. One of the more interesting results of the study was that errors of the first kind have about the same magnitude as errors of the second kind.

---

* Single precision has been selectively added to the CDC version of NASTRAN for most matrix operations. Details are described in Section 5.5.1 of the Programmer's Manual.

ERROR ANALYSIS

The equilibrium equation for the displacement method

$$[K]\{u\} = \{P\} \quad , \tag{1}$$

becomes, in the presence of errors in the stiffness matrix,

$$[K + K_\varepsilon]\{u + u_\varepsilon\} = \{P\} \quad , \tag{2}$$

where $[K_\varepsilon]$ is the error in the stiffness matrix and $\{u_\varepsilon\}$ is the resulting error in the solution vector. Neglecting the product of two errors and subtracting Equation 1 from Equation 2 results in

$$[K]\{u_\varepsilon\} = -[K_\varepsilon]\{u\} \equiv \{P_\varepsilon\} \quad . \tag{3}$$

This result shows that the error vector can be (approximately) calculated from the application of a load vector, $\{P_\varepsilon\}$, to the original structural system which is the product of the error matrix and the correct displacement solution vector.

An idea of the magnitude of error due to this source in uniform beams may readily be constructed if it is assumed that the error exists only in the diagonal (self) term for displacement at the free end of an end-loaded cantilever beam. Let the magnitude of the error be one binary bit in the 27th place. The term under consideration in the stiffness matrix is equal to $12EI/\Delta x^3$ where Δx is the length of one element in the beam. Thus the corresponding term in the error matrix is

$$K_\varepsilon = 2^{-27} \cdot \left(\frac{12EI}{\Delta x^3}\right) \quad . \tag{4}$$

The correct solution for the displacement at the free end is

$$u = \frac{P\ell^3}{3EI} \quad , \tag{5}$$

where ℓ is the total length of the beam. The load to be used in Equation 3 (which in this case reduces to a scalar equation) is, therefore,

$$P_\varepsilon = -K_\varepsilon u = -2^{-27} \cdot 4P\left(\frac{\ell}{\Delta x}\right)^3 \quad . \tag{6}$$

The ratio of the displacement error to the true displacement at the free end is equal to the ratio of load, so that

$$\frac{u_\varepsilon}{u} = \frac{P_\varepsilon}{P} = -2^{-27} \cdot 4N^3 \quad , \tag{7}$$

where $N = \ell/\Delta x$ is the number of cells in the cantilever beam. This result is shown in Figure 1 for various numbers of elements in the beam as the line labeled "tip modified."

A more meaningful (and probably conservative) estimate for the error ratio can be calculated by assuming that all diagonal elements in the stiffness matrix are in error in the same direction by one binary bit in the 27th place. The result of this calculation for a uniform cantilever beam and for uniform square frames is shown in Figure 1 as the lines labeled "all elements modified." It will be noted, in the case of the beam, that the error increases as the fourth power of the number of grid points, as may be expected since the number of terms in the error load vector, $\{P_\varepsilon\}$, increases in direct proportion to the number of grid points, and the error in displacement for a single term increases as N^3. If the errors are randomly distributed with respect to sign, the ratio of the net error loading to the error loading for a single term will increase approximately in proportion to the square root of the number of grid points. Thus it may be expected that, in practical problems where rounding errors are randomly distributed, the error in the displacement at the tip of the beam will increase as the 3.5 power of the number of grid points.

Several experimental points indicating the errors for actual problems are also shown in Figure 1. In the experiments, the accumulation of significant additional error during triangular decomposition was avoided by using higher precision in the decomposition. Severe local variations in the stiffness of elements can cause substantially greater errors than those indicated in Figure 1.

Examination of Figure 1 indicates that if three significant decimal digits are desired in the solution vector and single-precision arithmetic is used to generate the stiffness matrix, cantilever beam problems are limited to about 20 grid points and square frames are limited to about 150 grid points. Since it was desired to solve larger problems, and since this error cannot in general be reduced substantially except by carrying more binary bits in the stiffness matrix, it was concluded that all stiffness matrices must be generated using double-precision arithmetic. With 8 additional decimal digits in the stiffness matrix, examination of Figure 1 indicates that the problem size for cantilever beams can be extended to several thousand grid points, and for square frames to hundreds of thousands of grid points before significant round-off error is accumulated.

The errors that occur during decomposition of the stiffness matrix (round-off errors) were investigated by experimental means only. Care was taken to ensure that the terms in the stiffness matrix were integers that would not be truncated in the conversion from decimal to binary arithmetic. Some experimental determinations of error ratios due to round-off in the solutions for cantilever beams and square frames are indicated in Figure 2. The results indicate that if three-figure accuracy is desired, the number of grid points is limited, with single-precision arithmetic, to about 30 for both cantilever beams and square frames. If, on the other hand, double-precision arithmetic is used, extrapolation of Figure 2 indicates that problem size can be extended to several thousand grid points for cantilever beams and to hundreds of thousands of grid points for square frames.

Some improvement in the accuracy of single precision calculation can be achieved by means of an iterative solution technique (see Section 3.6.2) provided that double precision is used in accumulating the residual load vectors. The error ratio must, however, be less than unity in order for the iteration algorithm to be stable, so that, by inspection of Figure 2, only a modest increase in problem size is afforded.

Similar conclusions on the necessity for using double-precision arithmetic for generating the stiffness matrix and for decomposing matrices result if one considers dynamic problems. Consider, for example, the fundamental frequency of a cantilever beam with a concentrated mass at the free end. Let the error in the stiffness matrix be equal to 2^{-27} in the self term for lateral displacement of the free end and be equal to zero elsewhere. Then the calculated frequency, is

$$\omega_c = \left(\frac{K + K_\varepsilon}{M}\right)^{1/2} , \tag{8}$$

where

$$K = EI/3\ell^3 ,$$

$$K_\varepsilon = 2^{-27} \cdot \frac{12EI}{\Delta x^3}$$

The ratio of the calculated frequency to the analytically correct frequency, ω_a, is

$$\frac{\omega_c}{\omega_a} = \left(\frac{K + K_\varepsilon}{K}\right)^{1/2} \simeq 1 + 1/2\,\frac{K_\varepsilon}{K} = 1 + 2^{-27} \cdot 2N^3 . \tag{9}$$

Thus the error in frequency is about one-half as great as the error in displacement for the corresponding static problem, Equation 7.

In the analysis of free-free structures the accumulation of truncation error in the formation of the stiffness matrix may be interpreted as a set of springs (either positive or negative) that restrain the rigid body motions. Analysis shows that, for direct numerical integration of the equations of motion, double-precision formation of the stiffness matrix is required in order to contain the errors in the rigid body motions. For such problems single-precision arithmetic is inadequate for beam problems with more than about ten grid points.

A consideration that does not enter into the examples treated above is the effect of variations in stiffness with position in the structure. A common mistake made by beginners in the displacement method is to simulate a rigid constraint by connecting two grid points with a very stiff spring. Such an element will, of course, obliterate the contributions of other elements to the subpartition of the stiffness matrix corresponding to the pair of grid points, and wrong answers will result. (A stiff element connected from an independent grid point to ground will not, on the other hand, produce errors.)

NASTRAN provides the means, via multipoint constraints and rigid elements, to avoid errors associated with extremely stiff elements. The user must, however, still decide whether to treat a stiff element as an elastic body or as a rigid body. The decision will probably be based on an estimate of whether the modeling error made in replacing an elastic element by a rigid connection is significant. Assuming that the elastic representation will be retained provided that the modeling error is greater that 10^{-3}, the expected degradation in accuracy due to increased round-off error is approximately a factor of 1,000. Even so, Figures 1 and 2 indicate that, if double-precision is used, accurate results can be obtained for cantilever beams with several hundred grid points and square frames with several thousand grid points.

The conclusion of the investigation can be stated very simply. For problems of the size for which NASTRAN has been designed, double-precision arithmetic (16 decimal digits) is necessary, and the accuracy of the solutions obtained with double precision arithmetic should be satisfactory, in the vast majority of cases.

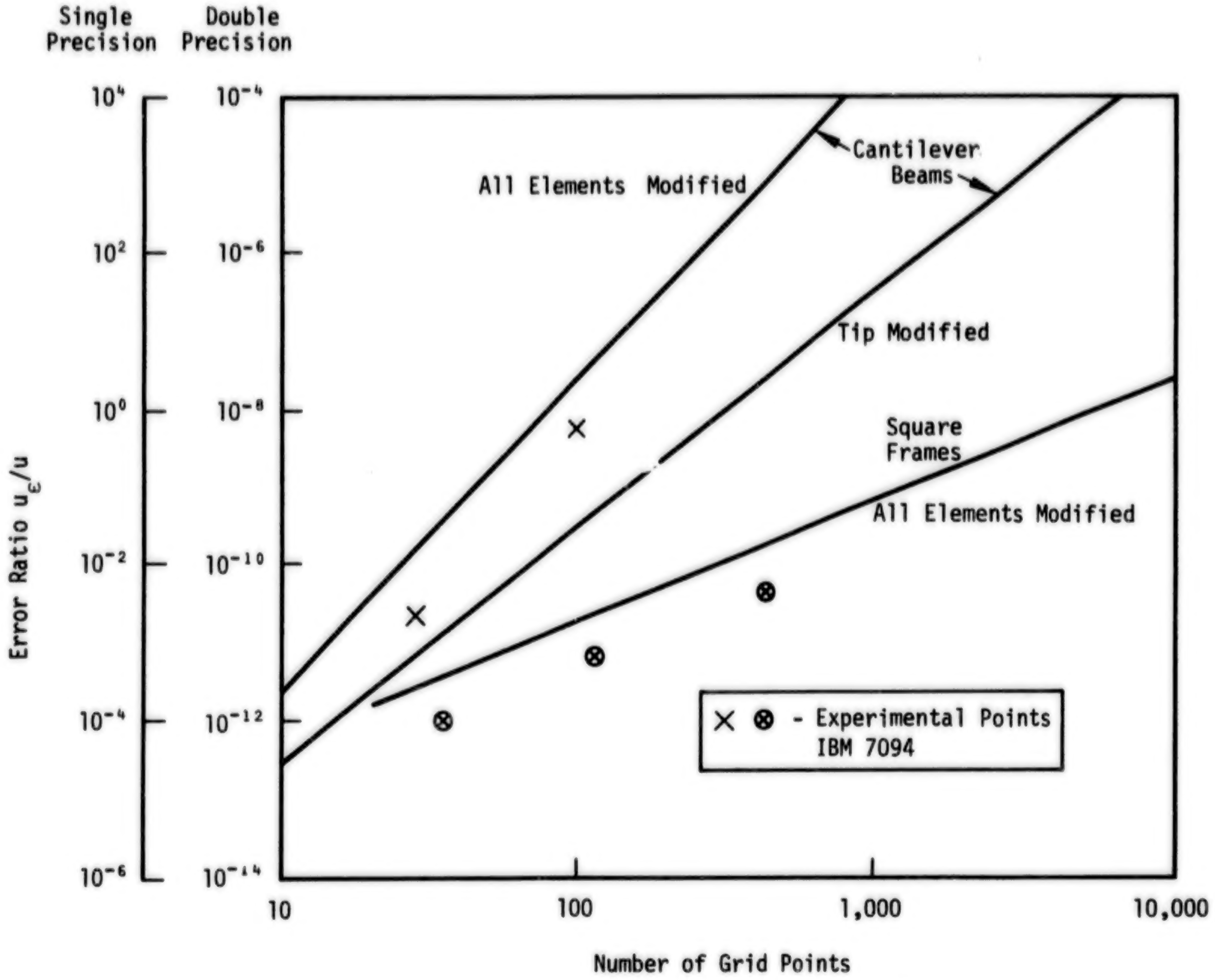


Figure 1. Effect of stiffness matrix error in static solutions.

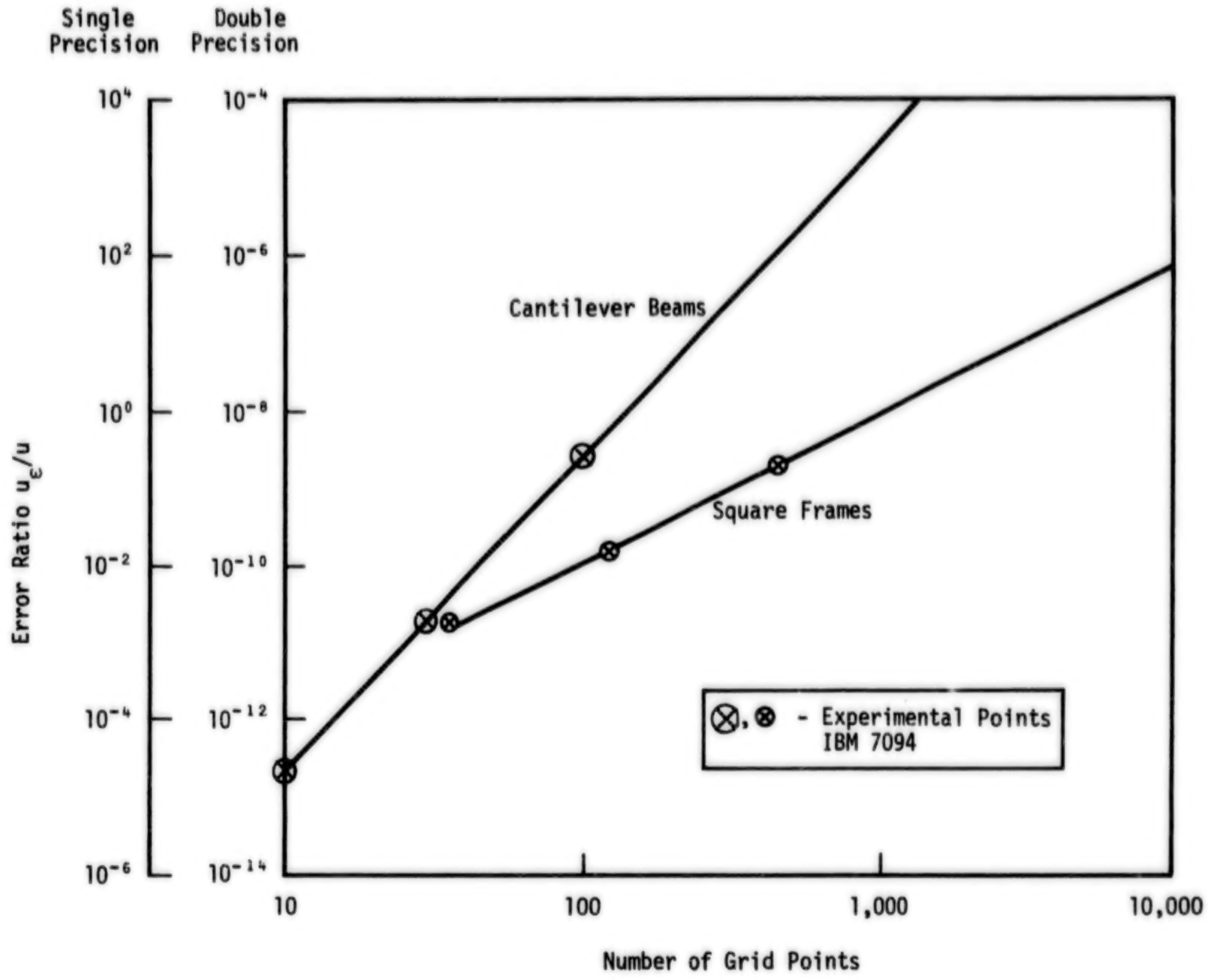


Figure 2. Effect of roundoff error in static solutions.

15.2 MODELING ERRORS IN THE BENDING OF PLATE STRUCTURES

15.2.1 Triangular Element

The errors associated with the use of the Clough bending triangle are documented in Reference 1. One of the checks made on the Clough triangle in NASTRAN was to run several plate bending problems and compare the results with those given in Reference 1. These results confirmed the equivalence between the triangle of Reference 1 and the triangular bending element in NASTRAN.

15.2.2 Quadrilateral Element

In order to check the accuracy of the QUAD1 element, comparisons were made with the Clough triangle and the Melosh rectangle (Reference 2). In developing the stiffness matrix for the Melosh rectangle, the bending curvatures are obtained by assuming that the displacements along the edges are third order polynomials, and that the curvature varies linearly in the direction normal to the edge. It is further assumed that the twisting effects are resisted by a state of uniform twist in the element. Unlike the NASTRAN quadrilateral, the Melosh rectangle cannot be applied to nonrectangular shapes.

Results of the comparisons of the NASTRAN quadrilateral with the Clough triangle and the Melosh rectangle are shown in Figures 2 through 9. The schedule of the eight cases considered is shown in Figure 1. All calculations were made on a quarter section of the plate with the indicated mesh sizes. The central deflection was used as a measure of the quality of the results. In the case of uniformly loaded plates, the central deflection is given by

$$w_c = \alpha \frac{q\,a^4}{D} \quad , \tag{1}$$

where α is a deflection coefficient, D is the flexural rigidity, a is the length of the shorter side (see Figure 1) and q is the load intensity.

For the concentrated load cases, the central deflection is given by

$$w_c = \beta \frac{P\,a^2}{D} \quad , \tag{2}$$

where β is a deflection coefficient and P is the concentrated load.

PAGE 869 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

Examination of the curves reveals that, in the case of the square plates, the NASTRAN quadrilateral gives results that are quite similar to those for the Melosh rectangle. However, for the plates with 2/1 aspect ratio, the Melosh rectangle gives somewhat better results, particularly for the coarse meshes. In all cases the quadrilateral gives better results than the triangle, particularly in the case of the square plates, where the accuracy is substantially better. The comparisons indicate that the NASTRAN quadrilateral is a good element to use for plate structures in the case of nearly rectangular geometry and with the aspect ratio near unity. This is precisely the case for which the element was designed.

Experience indicates that the error associated with increased aspect ratios tends to grow roughly in proportion to the aspect ratio. Therefore, it is desirable to avoid quadrilateral elements with large aspect ratios, particularly when using coarse meshes. Although the quadrilateral element is not restricted to a plane figure, it should not be used in modeling sharply curved surfaces. In this case triangular elements are preferred.

Triangles are also preferred if the geometry departs significantly from a rectangular pattern. Experience with rhombic elements, having a 45-degree sweep angle, indicates errors that are two to five times greater when using quadrilateral elements than when using triangular elements with acute angles. The error is even greater if obtuse triangular elements are used. The large error for the quadrilateral element in the 45° swept case is probably the result of the composite nature of the element. Relatively large errors are associated with the two obtuse triangles forming half of the quadrilateral element, and significantly smaller errors are associated with the two acute triangles forming the other half of the element, see Section 5.8.3.2.

With regard to twisting behavior, the quadrilateral element does not involve any approximation for the case of a square element subjected to uniform twisting moments along its edges. In this case, the constraints imposed on the diagonals of the quadrilateral conform exactly to the deformed shape.

15.2.3 Higher Order Triangular Plate Bending Element

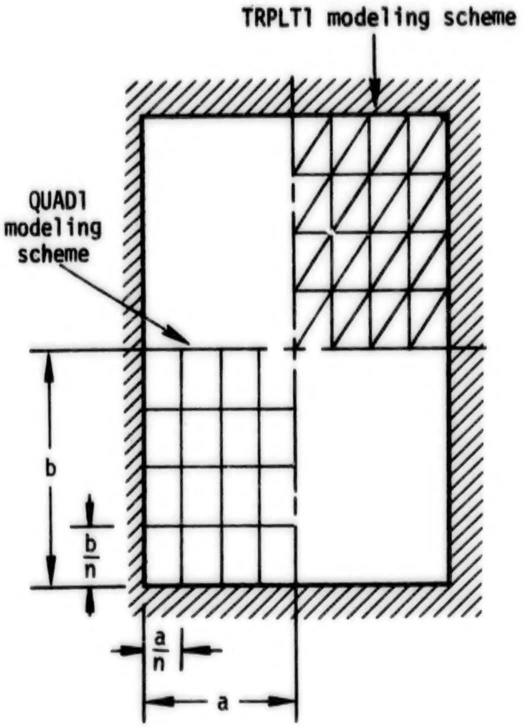
The rectangular plate shown in Figure 1 was also analyzed by using the higher order triangular plate bending element (TRPLT1) employing the modeling scheme shown in the figure. Because of symmetry, all calculations were made on only a quarter section of the plate with the indicated mesh sizes.

The results of the analyses are also presented in Figures 2 through 9. As can be seen, the results with the TRPLT1 element are more accurate in some cases than those with the QUAD1 element. This is due to the use of the quintic displacement field employed in the formulation of this element.

REFERENCES

1. Clough, R. W., and Tocher, J. L., "Finite Element Stiffness Matrices for Analysis of Plate Bending," Proc. of Conference on Matrix Methods in Structural Mechanics, Air Force Flight Dynamics Laboratory, Report AFFDL-TR-66-80, December 1965.

2. Melosh, R. J., "A Stiffness Matrix for the Analysis of Thin Plates in Bending," Journal of Aeronautical Sciences, Vol. 28, 1961, p. 34.


TRPLT1 modeling scheme
QUAD1 modeling scheme
b
b/n
a/n
a

CASE	b/a	EDGES	LOAD
1	1	SS	U
2	2	SS	U
3	1	C	U
4	2	C	U
5	1	SS	C
6	2	SS	C
7	1	C	C
8	2	C	C

Edges: Simply Supported (SS) or Clamped (C)
Loading: Uniform (U) or Concentrated at Center (C)

Figure 1. Schedule of rectangular plate analyses.

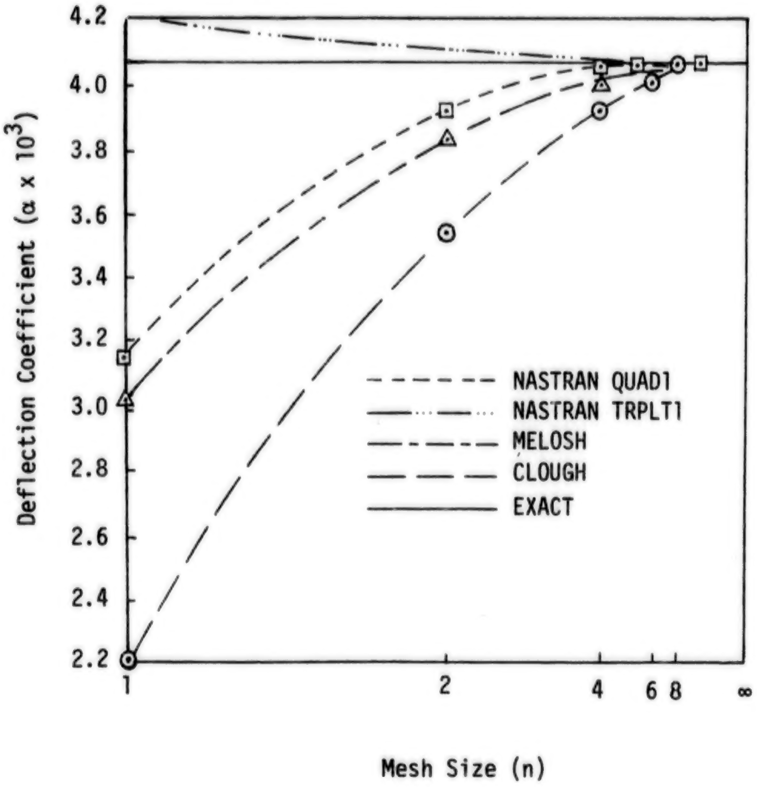


Figure 2. Central deflection of rectangular plate. Case 1 (1-SS-U)

875

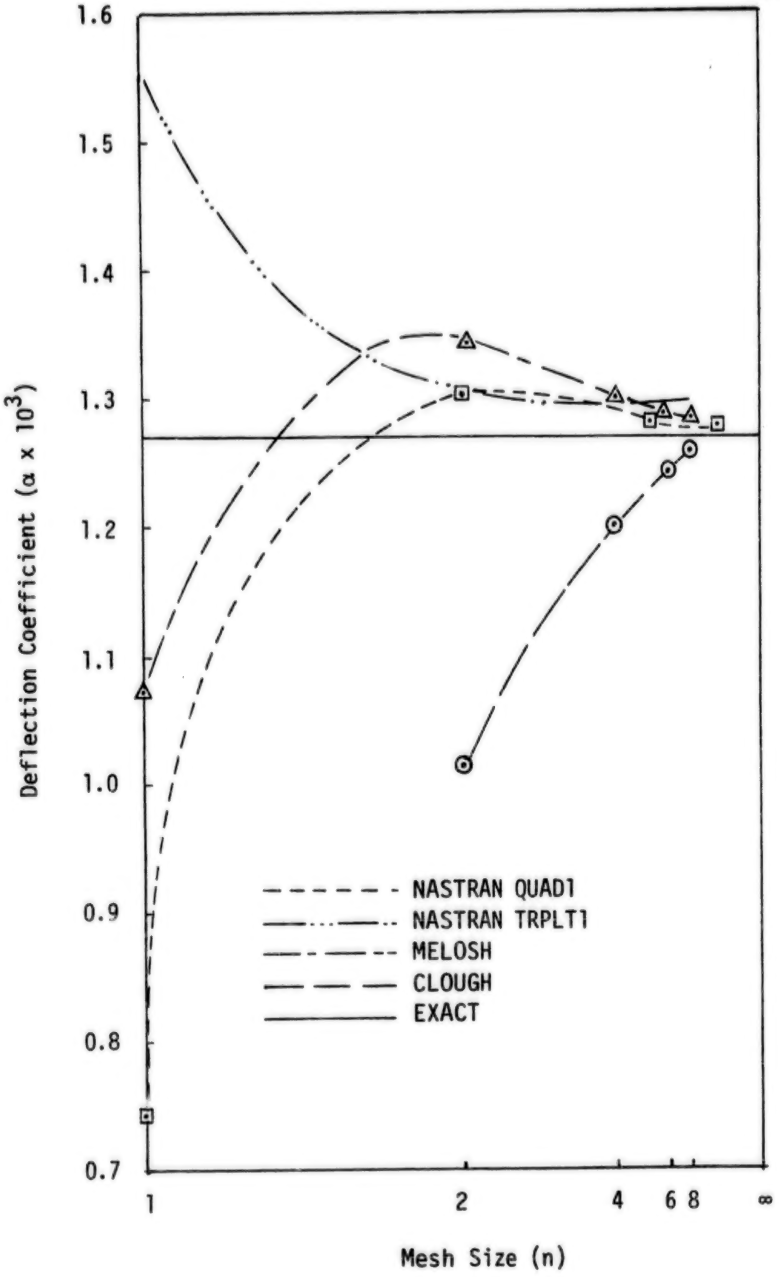


Figure 3. Central deflection of rectangular plate. Case 3 (1-C-U)

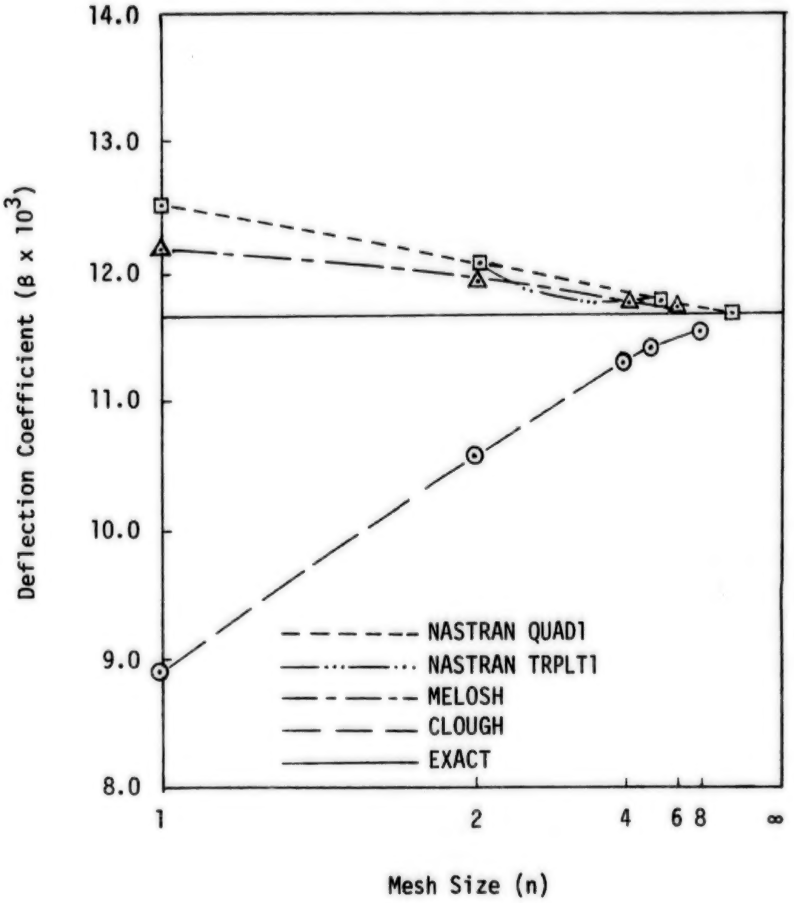


Figure 4. Central deflection of rectangular plate. Case 5 (1-SS-C)

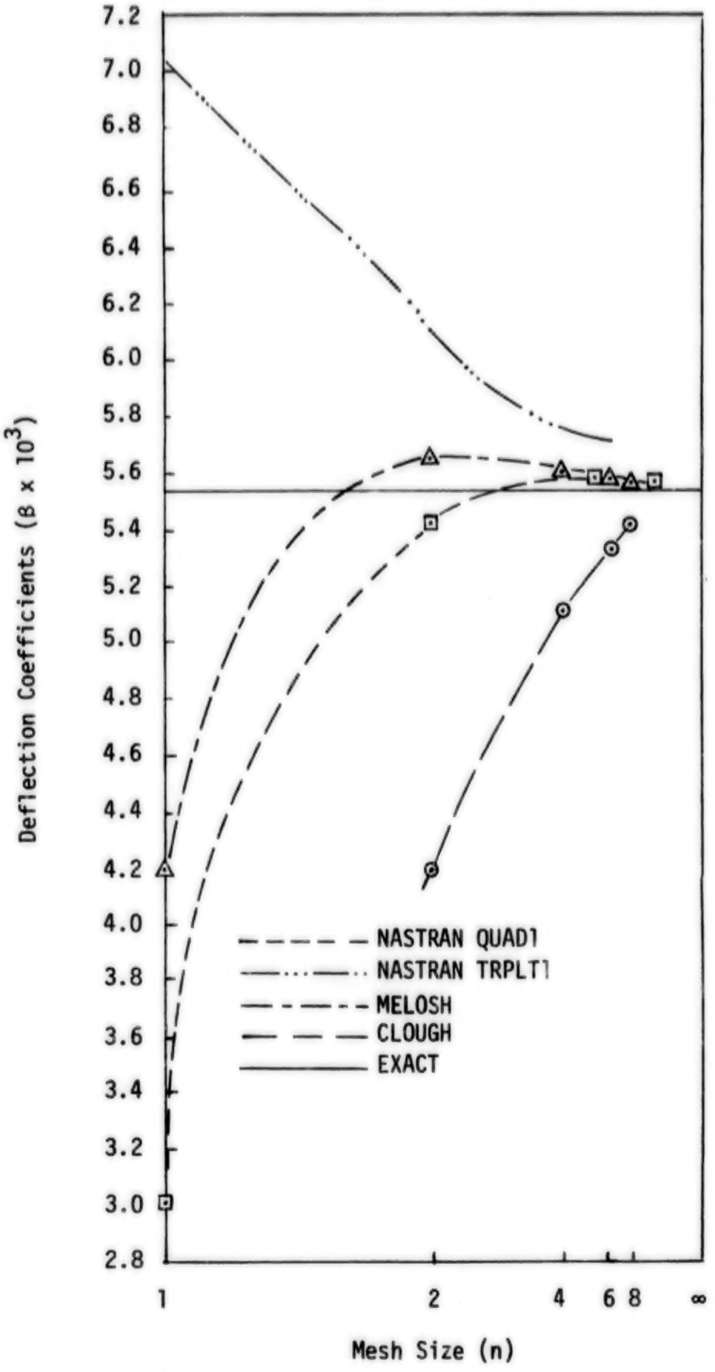


Figure 5. Central deflection of rectangular plate. Case 7 (1-C-C)

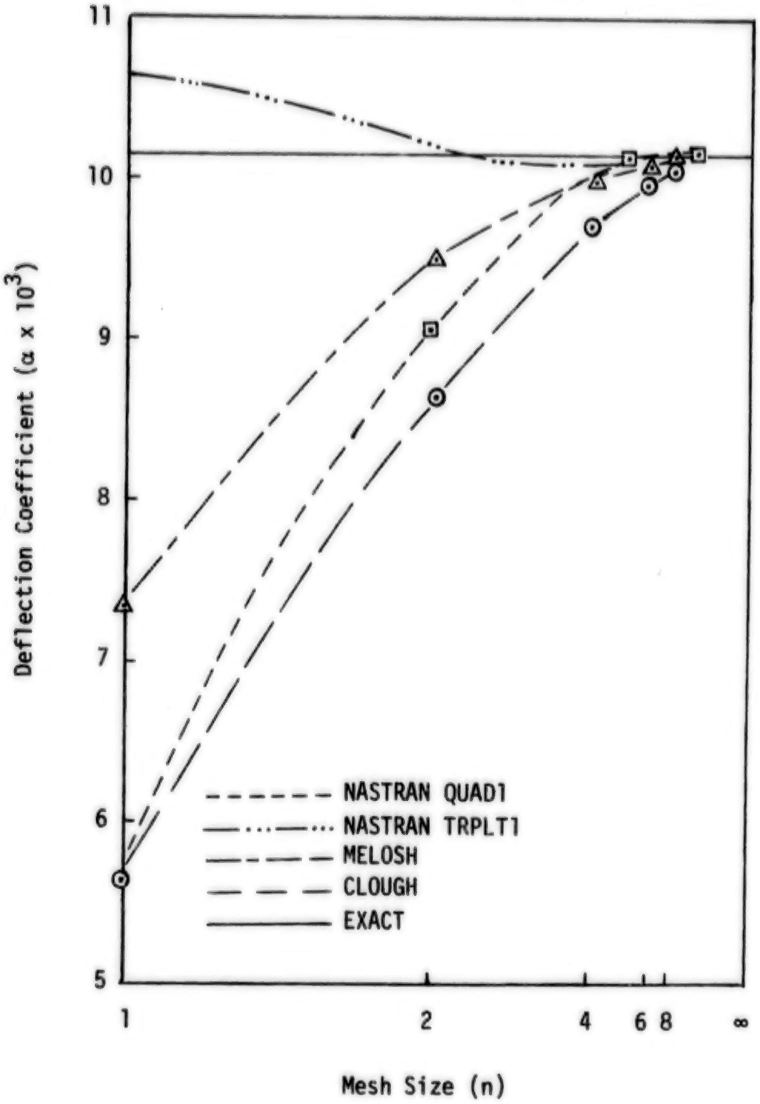


Figure 6. Central deflection of rectangular plate. Case 2 (2-SS-U)

879

ORIGINAL PAGE IS
OF POOR QUALITY

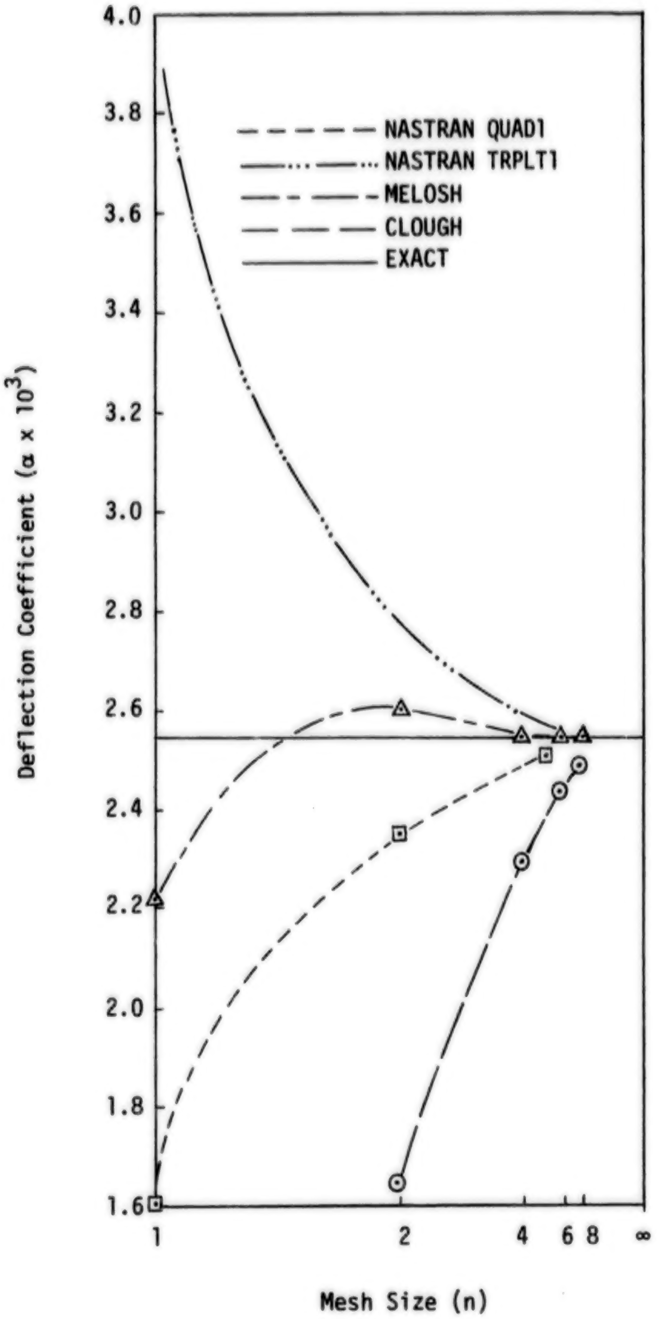


Figure 7. Central deflection of rectangular plate. Case 4 (2-C-U)

880

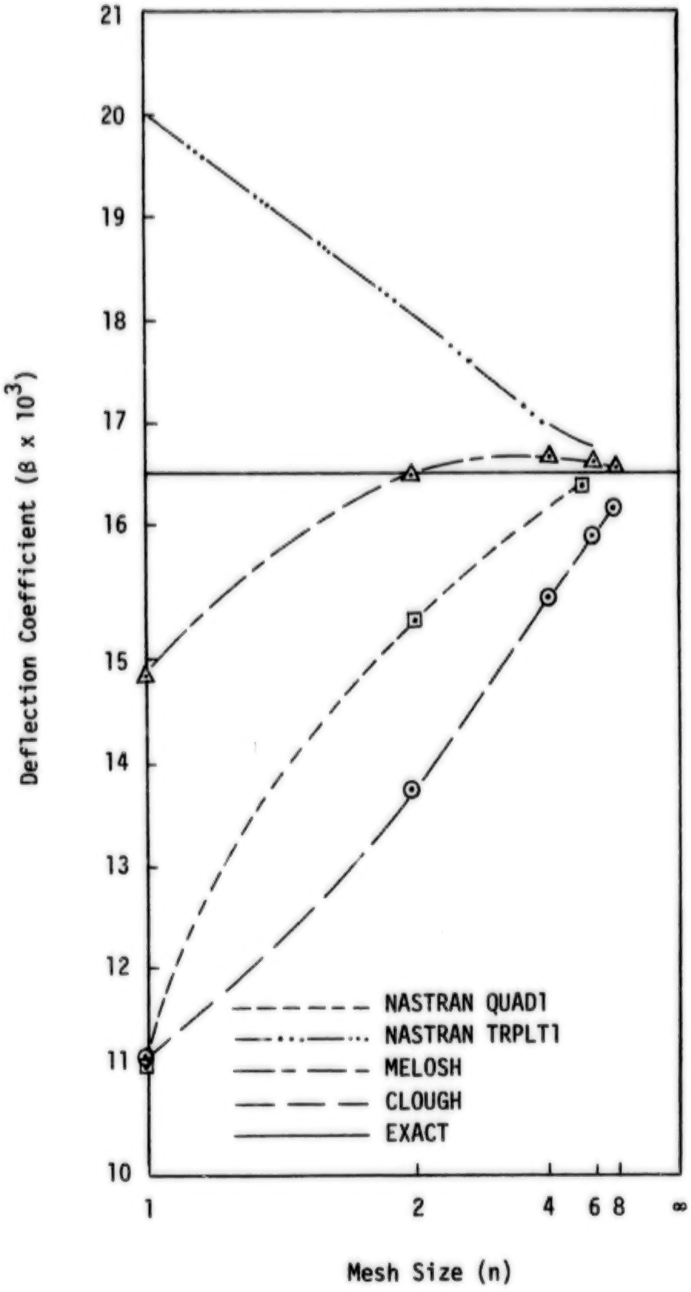


Figure 8. Central deflection of rectangular plate. Case 6 (2-SS-C)

ORIGINAL PAGE IS
OF POOR QUALITY

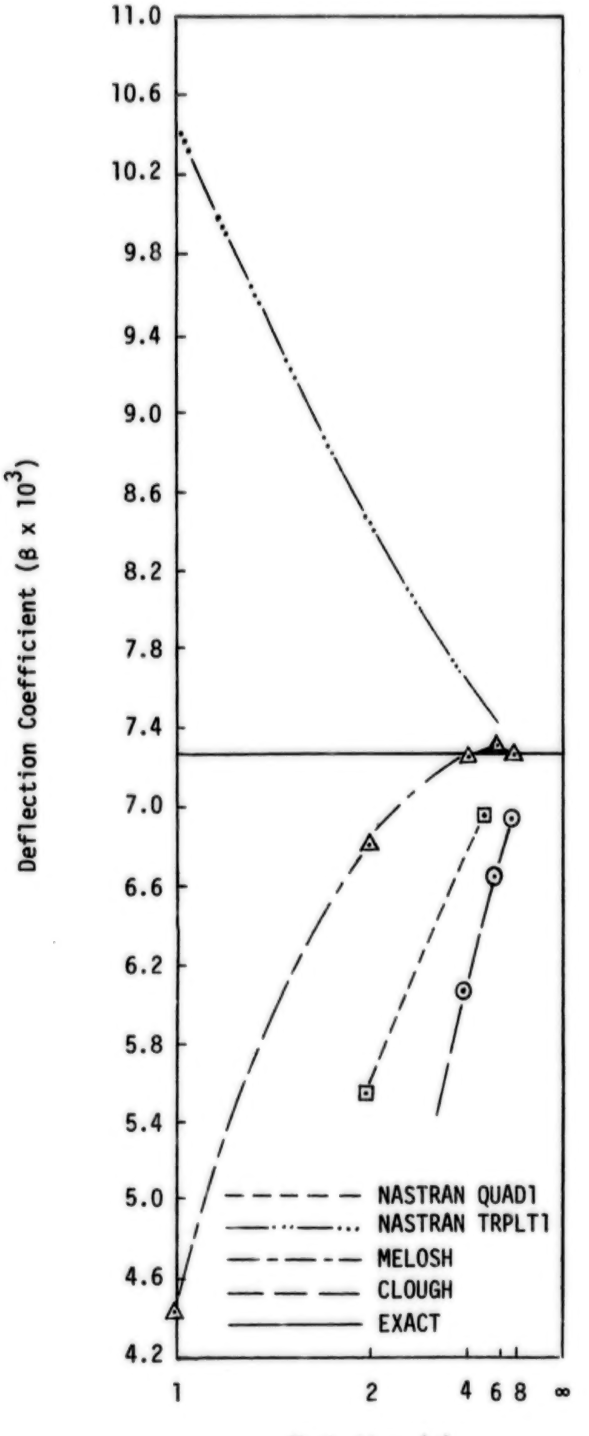


Figure 9. Central deflection of rectangular plate. Case 8 (2-C-C)

882

15.3 MODELING ERRORS IN MEMBRANE ELEMENTS

15.3.1 Quadrilateral Membrane Elements

NASTRAN includes three quadrilateral membrane elements (QDMEM, QDMEM1, and QDMEM2) which are identical in their connections and physical properties but which produce different results due to differences in mathematical formulation. All three elements give identical correct results for a flat element in a state of uniform plane stress. The differences occur when the element is warped out of plane and, more significantly, when the state of stress is not uniform within the element.

A frequent application of quadrilateral membrane elements is to model the central web of a beam, in which case they are subjected to a steep constant stress gradient. The extraordinarily large errors displayed in this application by the original NASTRAN quadrilateral membrane element (QDMEM) were an important motivating force for introducing the other two quadrilateral membrane elements. Figure 1 shows a comparison of results for the three NASTRAN elements when a pure bending couple is applied to a square. The improvement of QDMEM1 and QDMEM2 over QDMEM is evident, although even their accuracies are low. The use of two triangular membrane elements (TRMEM) will produce almost the same result as QDMEM in this example. In the case of the isoparametric element (QDMEM1), the result shown in Figure 1 can be extended to a general rectangular shape by means of the formula

$$\frac{\Delta\theta_{\text{QDMEM1}}}{\Delta\theta_{\substack{\text{BEAM}\\\text{THEORY}}}} = \frac{1-\nu^2}{1 + \frac{1-\nu}{2}\left(\frac{\ell}{h}\right)^2} \tag{1}$$

Better accuracy can be achieved by subdividing the beam web into two or more rows of elements. Figure 2 shows results taken from Reference 1 for a cantilever beam modeled by two rows of elements. The error in tip displacement for QDMEM1 is 14%, which might be acceptable for some applications, but the error for QDMEM is still unacceptable by any standard.

15.3.2 Linear Strain Triangular Membrane Element

The cantilever beam of Figure 2 is modeled by using eight equal linear strain triangular membrane elements (TRIM6). The modeling scheme and the results are shown in Figure 3 (in which the results of Figure 2 are reproduced to facilitate comparison). The high accuracy obtained from the use of the TRIM6 element is quite evident from this figure. This is due to the use of a quadratic displacement polynomial in the formulation of this element.

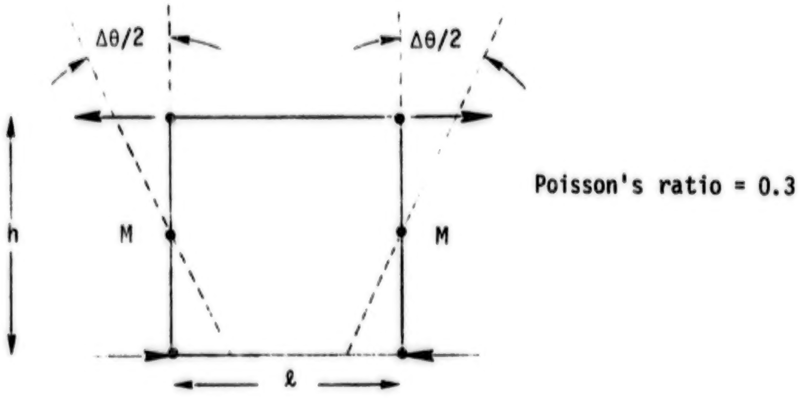
PAGE 883 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

ERROR ANALYSIS

REFERENCE

1. Adelman, H. M., Walz, J. E., and Rogers, J. L., Jr., "An Isoparametric Quadrilateral Membrane Element for NASTRAN," NASTRAN: Users' Experiences, NASA TM X-2637, September 1972, pp. 315-336.


Δθ/2
Δθ/2
h
M
M
ℓ
Poisson's ratio = 0.3

$\frac{\Delta\theta/M}{\ell/EI}$	Beam Theory	QDMEM	QDMEM1	QDMEM2
	1.000	.224	.675	.585

Figure 1. Comparison of flexibilities for a pure bending couple applied to square membrane elements, $\ell = h$

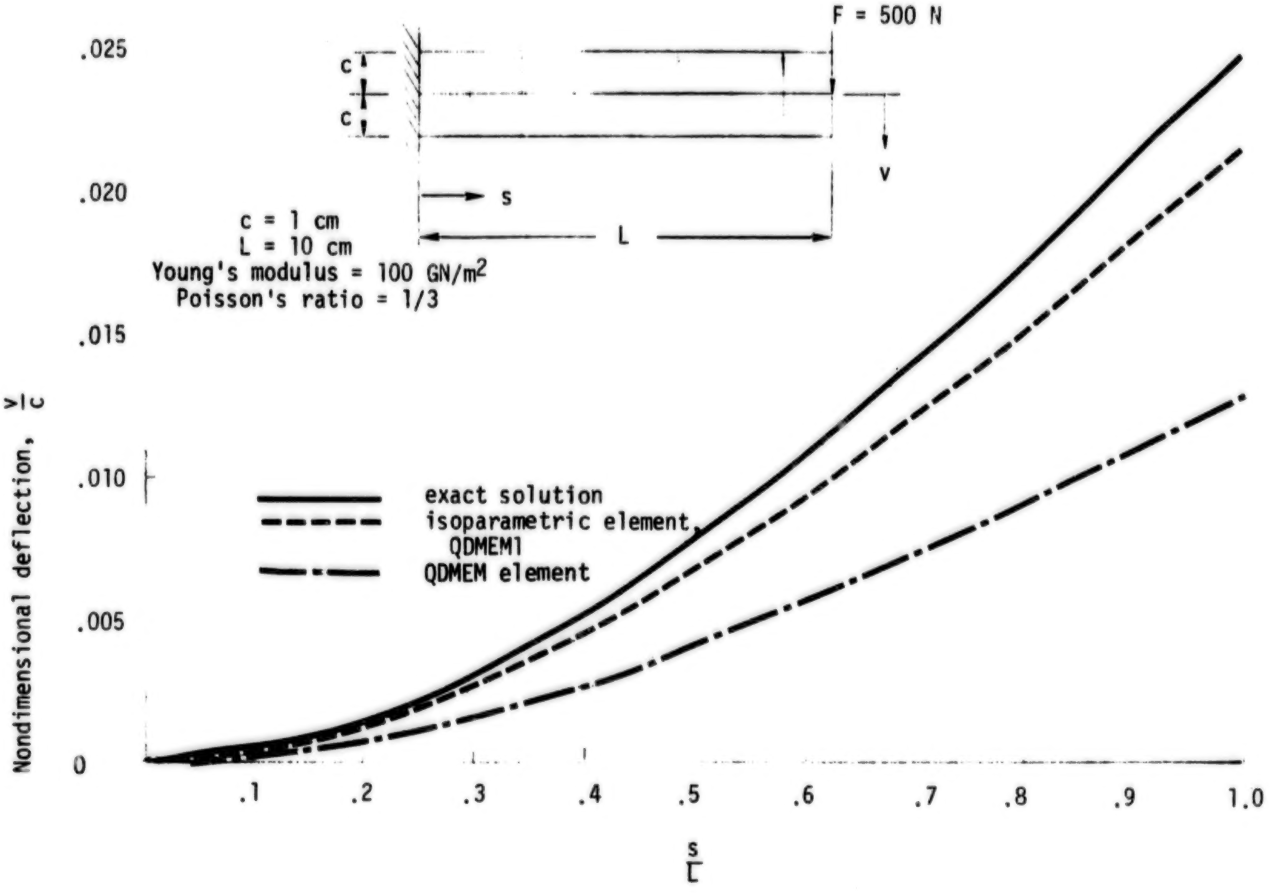
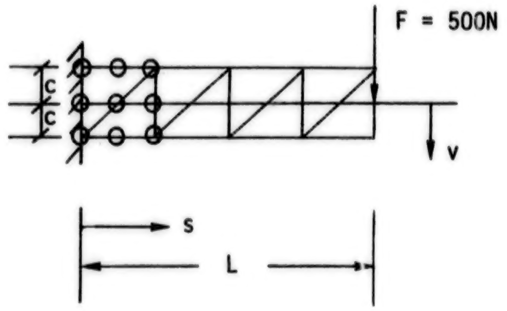


Figure 2. Deflection of cantilever beam idealized by quadrilateral membrane elements



TRIM6 modeling scheme

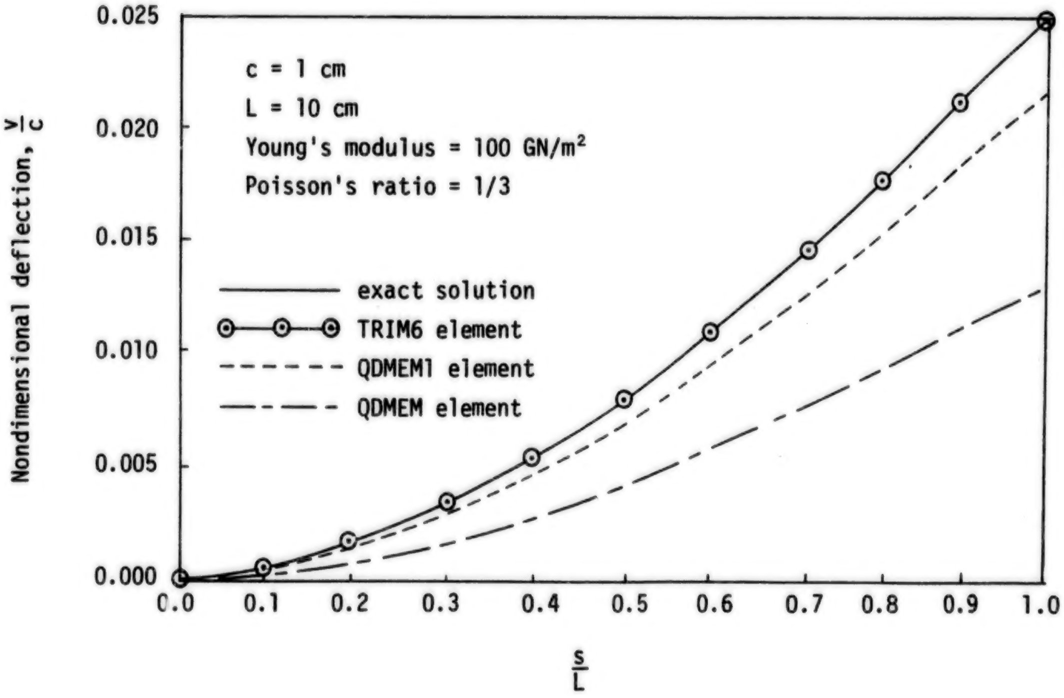


Figure 3. Deflection of cantilever beam idealized by QDMEM, QDMEM1, and TRIM6 elements

15.4 DECOMPOSITION ERROR CHECKS

The question of numerical accuracy is a constantly recurring issue. The sources for such errors can be attributed to the discretization of the original model, grid point sequencing schemes, and numerical manipulation errors. Once the model has been established, the solution accuracy comes into question. The causes of questionable results can sometimes be determined by the magnitudes and signs of the diagonal matrix terms which result during the decomposition of the stiffness matrix.

A special decomposition routine, SDCMPS, provides the following when erroneous matrix terms are detected:

1. Singular columns. Prior to actual decomposition, null columns are detected and the actual decomposition is not attempted. This is usually caused when an unconnected degree of freedom is not constrained by a single-point or multipoint constraint or by a rigid element.
2. Positive definiteness check. Negative or zero diagonal entries of the decomposed matrix are detected and listed. User options are provided to terminate after decomposition depending on the type and number of potential errors. Zero diagonal entries have 1.0 substituted to prevent numerical problems. Absolute values of negative entries are similarly substituted only if the Choleski option is being exercised.
3. Singularity error measure. The round-off error is estimated by a ratio of the decomposed and input diagonal terms and a measure of the word length, using the equation:

$$e_s = \frac{2^{1-P}}{|D_{ii}/K_{ii}|} , \qquad (1)$$

where P is the number of bits in the mantissa of real values on the machine being used, D_{ii} is the decomposed diagonal entry, and K_{ii} is the input diagonal entry. This is compared to the user's requested accuracy criteria, T_s, where $T_s = 2^{-n}$ and n is the number of bits of accuracy desired. Note that n should be less than 24, 27, or 48 for single precision words on IBM, UNIVAC, or CDC machines respectively. Values of e_s exceeding T_s suggest near singularities.

PAGE 889 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

4. Nonconservative matrix. If the decomposed diagonal term is larger than the original input diagonal term, the matrix is nonconservative. This may occur if negative springs are introduced into the matrix.

These matrix checks are particularly significant to real symmetric decomposition for static analyses. They do not retain this significance for eigenvalue extraction, unsymmetric matrix decomposition, or complex matrix decomposition.

For user convenience, the column numbers referenced in the diagnostic output are converted to grid point/component format. User options are provided for control over continuing or aborting the decomposition.

16. INTERACTION BETWEEN STRUCTURES AND FLUIDS

16.1 COMPRESSIBLE FLUIDS IN AXISYMMETRIC TANKS

16.1.1 Approach

In the class of problems considered here, the motions of the fluid are assumed to be small compared to the dimensions of the container so that nonlinear terms in the equations of motion can be ignored. Such problems should properly be classified as acoustic problems rather than as fluid flow problems. Another general restriction is that the shape of the container is axisymmetric. There is, however, no implication that the structure of the container has axisymmetric mechanical properties or that the combined motions of the fluid and the structure are axisymmetric. The restriction regarding container shape is introduced in order to simplify the equations of the fluid which, due to the assumption of small motions, can then be decomposed into uncoupled Fourier components with respect to azimuth position.

Compressibility of the fluid and the effects of gravity on a free surface are both included in the formulation. Thus, the capability described below can be used to solve a wide practical range of fluid-structure interaction problems for both liquids and gases.

Two different mathematical approaches are practical for the solution of linear fluid-structure interaction problems with interface surfaces of arbitrary shape. In the first approach a fundamental solution of the fluid field equations is used to formulate an integral equation relating motions and pressures at the fluid surface; the integral equation is then replaced by a finite set of simultaneous linear equations. In the second approach the field equations are approximated by a set of linear difference equations (or linear finite element equations) in which the variables are defined at a finite set of points within the fluid and on its surface. The first approach has the advantages that it results in a smaller number of equations and that the user is not burdened with the task of locating grid points within the fluid. It has the disadvantages that the resulting matrices are completely full, that problems involving compressible fluids can only be formulated in the frequency domain, that the matrix coefficients are transcendental functions of frequency if the fluid is compressible, and that the effects of inhomogeneous fluid density cannot be treated. It is, however, an excellent method for homogeneous, incompressible fluids and it is the only practical method when the fluid is infinite in extent. The second approach has been selected for the class of problems considered here mainly because it can easily handle compressibility and space-variable fluid density for all types of analysis. The restriction to axisymmetric shapes reduces the disadvantage regarding the location of grid points in the fluid,

892

in that grid points need only be specified on a cross-section of the fluid and not throughout its volume. A finite element formulation has been selected, rather than a finite-difference formulation, in order to facilitate the solution of problems with arbitrary axisymmetric tank shapes.

The field equations can be expressed directly in terms of the components of motions or, if the fluid is irrotational, they can be reduced to the solution of the wave equation for a scalar potential function. The latter formulation has the advantage that the number of variables is only one third as large, but it has the disadvantage that the enforcement of compatibility at the boundary between the fluid and the structure is awkward. The awkwardness is not, however, so great as to cause serious difficulties in problem solution, so that a formulation in terms of a scalar potential function (specifically the pressure) has been adopted.

In summary, the selected approach is one in which the unknown variables include the harmonic components of pressure at a finite number of grid points in the fluid, and also the structural displacements throughout the structure, including points on the interface. The harmonic components of pressure are mathematically treated by NASTRAN as components of displacement. The inertia properties of the fluid are represented by a matrix that is treated like a stiffness matrix and the compressibility is represented by matrix terms similar to a mass matrix. These matrices are generated from the properties of "fluid elements" interconnecting fluid grid points. The coupling between the fluid and the structure is represented by special matrices that are treated as direct input matrices (see Section 9.3.3).

The above capability is implemented in NASTRAN primarily by modification of existing functional modules. The modules that are affected most significantly are the Input File Processor, the Structural Matrix Assembler (which processes the fluid elements), and the Geometry Processor. A new functional module is introduced to generate the fluid-structure interaction matrices.

All of the existing analysis types are available for the solution of fluid-structure interaction problems, including the calculation of vibration modes, the solution of frequency response problems and the solution of transient problems. In addition, vibration modes can be obtained for the fluid alone, with a rigid container replacing the structure.

16.1.2 Fluid Field Equations

The governing equations used in the analysis of the irrotational motions of an inviscid compressible fluid are:

$$\rho_d\ddot{\vec{u}} + \frac{1}{2}\rho_d\nabla(\dot{\vec{u}}\cdot\dot{\vec{u}}) = -\nabla p - \rho_d\vec{g} \quad , \tag{1}$$

$$p = -B\nabla\cdot\vec{u} \quad , \tag{2}$$

$$\rho_d = \rho - \nabla\cdot(\rho\vec{u}) \quad , \tag{3}$$

where:

$\vec{u}$ is a small displacement of the fluid from a fixed reference point,

$\dot{\vec{u}}$ is the velocity, $(\partial\vec{u}/\partial t)$,

$\ddot{\vec{u}}$ is the acceleration, $(\partial^2\vec{u}/\partial t^2)$,

ρ is the mass density of the fluid, at rest,

ρ_d is the mass density of the perturbed fluid,

$\vec{g}$ is the vector of the gravity field (opposite to the direction of free fall),

p is the pressure,

B is the bulk modulus of elasticity.

Due to the assumption of small motions, the second term on the left side of Equation 1 will be ignored. The second term on the right side of Equation 3 will be ignored on the grounds that it is small and unimportant for most problems. If small motions are assumed, it is significant only in problems where the effects of gravity are included and the density varies throughout the fluid. With these assumptions, Equation 1 may be replaced by the following simplified equation:

$$\rho\ddot{\vec{u}} = -\nabla p - \rho\vec{g} \quad . \tag{1a}$$

Equations 1a and 2 are combined as follows to obtain a single equation in terms of the pressure. Taking the second time derivative of Equation 2, and the divergence $(\nabla\cdot)$ of Equation 1a we obtain:

$$\ddot{p} = -B\nabla\cdot\ddot{\vec{u}} \quad , \tag{4}$$

$$\nabla\cdot\ddot{\vec{u}} = -\nabla\cdot\frac{1}{\rho}\nabla p - \nabla\cdot\vec{g} \quad . \tag{5}$$

Combining Equations 4 and 5:

$$\frac{1}{B}\ddot{p} - \nabla\cdot\frac{1}{\rho}\nabla p = \nabla\cdot\vec{g} = 0 \quad , \tag{6}$$

where $\nabla\cdot\vec{g} = 0$ is assumed in the following development.

Equation 6 is a form of the wave equation. Energy expressions will be derived from it, with the aid of which equivalent stiffness and mass matrices can be formed. To this end define the sum of the terms on the left side of Equation 6 to be the generalized force;

$$J = \frac{1}{B}\ddot{p} - \nabla\cdot\frac{1}{\rho}\nabla p \quad . \tag{7}$$

Introducing the concepts of the calculus of variations, the volume integral of the product of the variation of the pressure, δp, times the generalized force will produce a variation of the generalized energy:

$$\delta H = \int_V J\,\delta p\,dV = \int_V \left(\frac{1}{B}\ddot{p} - \nabla\cdot\frac{1}{\rho}\nabla p\right)\delta p\,dV \quad . \tag{8}$$

The ∇ operations may be transformed by the identity:

$$\nabla\cdot\left[\delta p\left(\frac{1}{\rho}\nabla p\right)\right] = \delta p\nabla\cdot\left(\frac{1}{\rho}\nabla p\right) + \frac{1}{\rho}\nabla(\delta p)\cdot\nabla p \quad . \tag{9}$$

Substituting Equation 9 into Equation 8 results in :

$$\delta H = \int_V \frac{1}{B}\ddot{p}\,\delta p\,dV + \int_V \frac{1}{\rho}\nabla p\cdot\nabla(\delta p)\,dV - \int_V \nabla\cdot\left[\delta p\left(\frac{1}{\rho}\nabla p\right)\right]dV \quad . \tag{10}$$

Applying the divergence theorem to the third integral produces an integral over the surface S

$$\int_V \nabla\cdot\left[\delta p\left(\frac{1}{\rho}\nabla p\right)\right]dV = \int_S \delta p\left(\frac{1}{\rho}\nabla p\right)\cdot d\vec{S} \quad , \tag{11}$$

where the vector $\vec{S}$ is normal outward from the surface. The first and second integrals in Equation 10 may be modified by the identities:

$$\begin{aligned} \delta\left(\frac{\partial x}{\partial t}\right)^2 &= 2\frac{d}{dt}\left(\frac{\partial x}{\partial t}\right)\delta x = 2\ddot{x}\delta x \quad , \\ \delta(\nabla x\cdot\nabla x) &= 2\nabla x\cdot\nabla(\delta x) \quad , \end{aligned} \tag{12}$$

where x is any function.

The variation in energy δH after applying Equations 11 and 12 to Equation 10 is:

$$\delta H = \delta\left[\int_V \left(\frac{1}{2B}\dot{p}^2 + \frac{1}{2\rho}\nabla p\cdot\nabla p\right) dV\right] - \int_S \delta p\left(\frac{1}{\rho}\nabla p\right)\cdot d\vec{S} \quad . \tag{13}$$

The generalized energy internal to the fluid is the value of the first integral which, it is seen, contains a term similar in form to kinetic energy, and a term similar in form to potential energy. The energy flow through the boundary is given by the second integral.

In the finite element method of solution, a set of variables, p_i, equal to the values of p at specific points, is chosen and the volume is divided into subregions, called fluid elements, with vertices defined by the location of the variables. A "stiffness" matrix [K] is formed from the potential energy U by the equation:

$$K_{ij} = \frac{\partial^2 U}{\partial p_i\, \partial p_j} \tag{14}$$

The "equivalent" potential energy for each subregion is, from Equation 13,

$$U = \int_V \frac{1}{2\rho}\nabla p\cdot\nabla p \; dV \quad . \tag{15}$$

The pressure field for each subregion (fluid element) is dependent on the pressures p_i at its vertices.

The "equivalent" kinetic energy for a fluid element is, from Equation 13,

$$T = \int_V \frac{1}{2B}(\dot{p})^2 \; dV \quad . \tag{16}$$

A "mass" matrix [M] is formed from the kinetic energy by the equation

$$M_{ij} = \frac{\partial^2 T}{\partial \dot{p}_i\, \partial \dot{p}_j} \quad . \tag{17}$$

The set of simultaneous equations describing the pressures at discrete points may then be written in matrix form as

$$[M_{ij}]\{\ddot{p}_j\} + [K_{ij}]\{p_j\} = \{I_i\} , \qquad (18)$$

where $\{p_j\}$ is the vector of pressures at fluid points and $\{I_i\}$ is the vector of generalized forces imparted to the fluid. For the uniform gravitational field treated by NASTRAN, $\{I_i\}$ is null except at the boundaries.

16.1.3 Boundary Conditions

Two types of fluid boundaries will be defined. The first is a free surface, which is defined as an area with zero pressure relative to ambient pressure. Since the fluid pressure is measured at points fixed in space, the actual surface will be displaced and the pressure measured at the original position may not be zero. The second type is a structural boundary, through which the fluid may not diffuse, and at which it must have displacement compatability normal to the surface. Viscous boundary layer and rough surface friction effects depend on relative tangential velocities. These effects will be ignored for the present.

A fluid-to-fluid interface is a trival matter since nonuniform fluid density and bulk modulus are allowed. Except for gravity effects, the pressure and flow through these interfaces are continuous and compatible. When finite elements are used, the differential equations of motion are set up in each fluid element separately and each element may have a separate set of physical parameters.

From Equation 13, Section 16.1.2, the outward energy flow through the boundary is:

$$\delta H_{(\text{boundary})} = -\int_S \delta p\left(\frac{1}{\rho}\nabla p\right)\cdot d\vec{S} \quad , \tag{1}$$

showing that

$$I_{S_i} = \int_{S_i} \left(\frac{1}{\rho}\nabla p\right)\cdot d\vec{S} \quad , \tag{2}$$

is the generalized force acting on the pressure over an element of surface area S_i. From Equation 1a of Section 16.1.2,

$$\frac{1}{\rho}\nabla p = -\ddot{\vec{u}} - \vec{g} \quad , \tag{3}$$

so that

$$I_{S_i} = -\int_{S_i} (\ddot{\vec{u}} + \vec{g})\cdot d\vec{S} \quad . \tag{4}$$

It is interesting to note that at interior points the quantity

$$\vec{I} = \frac{1}{\rho}\nabla p = -\ddot{\vec{u}} - \vec{g} \quad . \tag{5}$$

assumes a role that is analogous to the gradient of the stress tensor in elasticity or to current density in electrical conduction.

On the structural boundary define

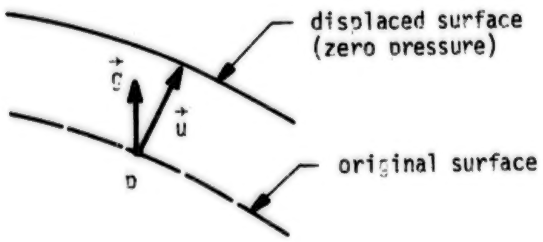
$$u_{ni} = \frac{1}{S_i}\int_{S_i} \vec{u}\cdot d\vec{S} \quad , \tag{6}$$

as the average normal displacement over S_i. Then from Equation 4

$$I_{S_i} = -S_i\,\ddot{u}_{ni} - \int_{S_i} \vec{g}\cdot d\vec{S} \quad . \tag{7}$$

At the structural interface, u_{ni} is a structural displacement. Equation 7 shows that the coefficient S_i has the form of a mass coefficient.

At a free surface the acceleration may be directly related to the pressure as follows.



For small motions it may be assumed that the pressure near the surface varies linearly with distance from the displaced surface. Thus, at a point on the original surface, using Equation 1a,

$$p = -\vec{u}\cdot\nabla p = \vec{u}\cdot\rho(\vec{g} + \ddot{\vec{u}}) \tag{8}$$

The pressure is evaluated at points on the original surface rather than on the displaced surface due to the Eulerian view point taken in fluid mechanics i.e., that the reference frame remains fixed as the fluid moves about. Differentiating Equation 8 twice with respect to time, and ignoring the nonlinear term, $\vec{u}\cdot\rho\ddot{\vec{u}}$,

$$\ddot{p} = \rho\ddot{\vec{u}}\cdot\vec{g} \tag{9}$$

It will be assumed that the gravity vector is in the direction of the outward normal to the original surface (which is necessary for the original surface to be in equilibrium). Thus

$$\ddot{\vec{u}}\cdot d\vec{S} = \frac{dS}{g}\ddot{\vec{u}}\cdot\vec{g} = \frac{dS}{\rho g}\ddot{p} \quad , \tag{10}$$

where dS and g are the magnitudes of $d\vec{S}$ and $\vec{g}$. Substituting Equation 10 into 4

$$I_{S_i} = -\int_{S_i} \frac{1}{\rho g}\ddot{p}\, dS - \int_{S_i} \vec{g}\cdot d\vec{S} \quad . \tag{11}$$

Define the average pressure over surface element S_i,

$$p_i = \frac{1}{S_i}\int_{S_i} p\, dS \quad . \tag{12}$$

Then from Equation 11, if ρg is uniform over S_i,

$$I_{S_i} = -\frac{S_i}{\rho g}\ddot{p}_i - \int_{S_i} \vec{g}\cdot d\vec{S} \tag{13}$$

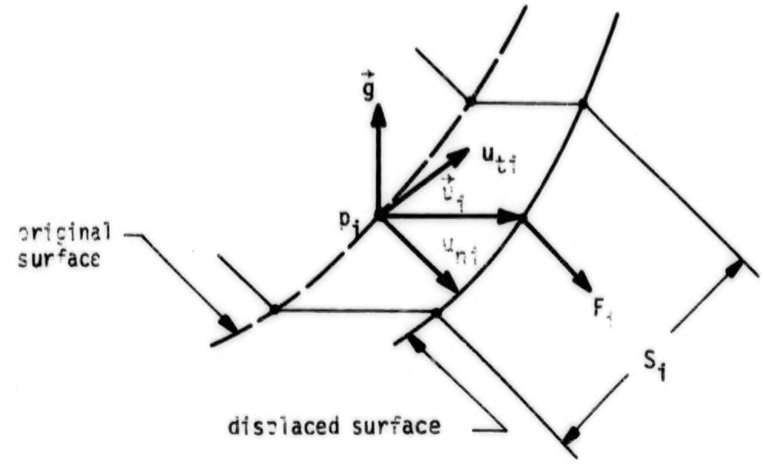
The first term in Equation 13 can be represented as a generalized inertia force in which the mass coefficient is $S_i/\rho g$. If the gravity field is absent (or is ignored) the mass coefficient becomes infinite and

$$p_i = 0 \quad , \tag{14}$$

on the surface. Equation 14 is treated in NASTRAN as a rigid constraint.

The effect of the term $\int_{S_i} \vec{g}\cdot d\vec{S}$ in Equations 7 and 13 is to produce a static variation of pressure with depth. It is neglected in NASTRAN dynamic analyses since pressures are measured relative to static equilibrium.

The force produced by the fluid on a structural grid point at the boundary is obtained as follows.



According to the Lagrangian formulation which is used in structural mechanics, the force on the structural interface is equal to the integral of the pressure over the displaced surface, which, to first order in small quantities, equals the pressure on the original surface plus a term due to the gravitational force on the fluid between the two boundaries. Thus the force on the boundary structural grid point in the direction of the normal is, assuming small motions,

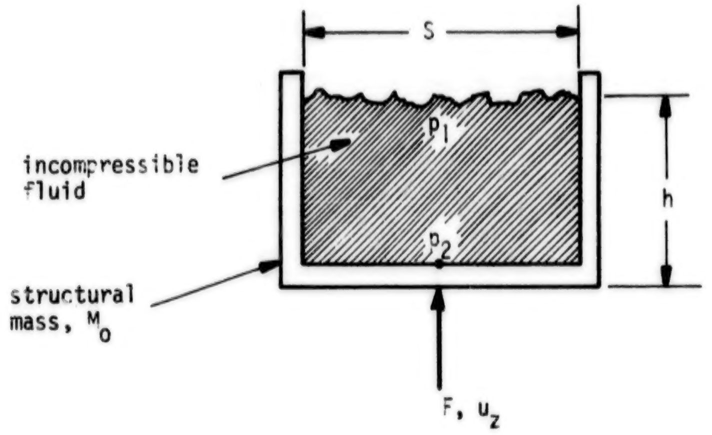
$$F_i = S_i(p_i - \rho\vec{g}\cdot\vec{u}_i) \quad , \tag{15}$$

where p_i is the average pressure defined in Equation 12 and $\vec{u}_i$ is the displacement vector of the structural point. The terms in Equation 7 and in Equation 15 can be written as follows in matrix form.

$$\left[\begin{array}{c} F_i \\ \hline I_{S_i} \end{array}\right] = \left[\begin{array}{c:c:c} -\rho g_{ti}\ S_i & -\rho g_{ni}\ S_i & S_i \\ \hdashline 0 & -S_i\ s^2 & 0 \end{array}\right] \left\{\begin{array}{c} u_{ti} \\ \hdashline u_{ni} \\ \hdashline p_i \end{array}\right\} \quad , \tag{16}$$

where s = d/dt and g_{ni} and g_{ti} are the components of gravity in the normal and tangential directions. Equation 16 is a statement of the boundary conditions at the structural interface.

As an example to illustrate the theory developed to this point consider the following simple problem.



Compute the vertical motion u_z, in response to an oscillating vertical applied force, F.

1. If uniform vertical motion is assumed, the pressure gradient is a constant function of the pressures, p_i, at the upper and lower surface:

$$\nabla p = \frac{\partial p}{\partial z} = \frac{p_1 - p_2}{h} \quad . \qquad (17)$$

From Equations 14 and 15, Section 16.1.2, the elements of the "stiffness" matrix are:

$$K_{ij} = \frac{\partial^2}{\partial p_i \partial p_j} \int_V \frac{1}{2\rho} (\nabla p)^2 \, dV$$

$$= \frac{\partial^2}{\partial p_i \partial p_j} \left[Sh \cdot \frac{1}{2\rho} \left(\frac{p_1 - p_2}{h} \right) \right]^2 \quad . \qquad (18)$$

The complete "stiffness" matrix is, therefore,

$$[K_{ij}] = \frac{S}{\rho h} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad . \qquad (19)$$

2. Due to free surface effects, the additional generalized force at point p_1 is, from Equation 13

$$I_{S_1} = - \frac{S}{\rho g} \ddot{p}_1 = - \frac{S}{\rho g} s^2 \, p_1 \quad . \qquad (20)$$

3. Noting the fact that the gravity vector and the displacement u_z are both opposite in direction to the normal vector at the lower fluid boundary, the forces due to the interface, obtained from Equation 16, are

$$\left\{\begin{array}{c} F_{z2} \\ \hdashline I_{s2} \end{array}\right\} = \left[\begin{array}{c:c} \rho g S & -S \\ \hdashline Ss^2 & 0 \end{array}\right]\left\{\begin{array}{c} u_z \\ \hdashline p_2 \end{array}\right\} , \tag{21}$$

where F_{z2} is the force on the structure, in the direction of u_z, due to fluid pressure.

4. The total equilibrium equation, including the contributions of Equations 19, 20 and 21, and of the structural mass, M_o, is:

$$\left\{\begin{array}{c} 0 \\ 0 \\ \Delta F \end{array}\right\} = \left[\begin{array}{c:c:c} \frac{S}{\rho h} + \frac{S}{\rho g} s^2 & -\frac{S}{\rho h} & 0 \\ \hdashline -\frac{S}{\rho h} & \frac{S}{\rho h} & -Ss^2 \\ \hdashline 0 & S & M_o s^2 - \rho g S \end{array}\right]\left\{\begin{array}{c} p_1 \\ p_2 \\ u_z \end{array}\right\} \tag{22}$$

The pressures and forces, it must be remembered, are measured relative to the static equilibrium. The applied force ΔF is therefore:

$$\Delta F = F - g(\rho Sh + M_o) \; . \tag{23}$$

The validity of Equation 22 is verified as follows. The first two rows of Equation 22 are solved to obtain:

$$p_1 = \frac{\frac{1}{h} p_2}{\frac{1}{h} + \frac{s^2}{g}} = \frac{g}{g + hs^2} \, p_2 \quad , \tag{24}$$

and

$$p_2 = p_1 + \rho h s^2 u_z \quad , \tag{25}$$

or, substituting Equation 24 into Equation 25,

$$p_2 - \frac{g}{g + hs^2} p_2 = \rho h s^2 u_z \quad ,$$

which reduces to

$$p_2 = \rho(g + hs^2)u_z \quad . \qquad (26)$$

Substituting Equation 26 into the third row of Equation 22 results in:

$$\Delta F = [\rho S(g + hs^2) + M_0 s^2 - \rho g S]u_z \quad .$$

Substituting for ΔF from Equation 23 produces the final equation of motion:

$$(M_0 + \rho h S)(\ddot{u}_z + g) = F \quad , \qquad (27)$$

which is the obvious answer for the uniform acceleration of the total mass of the system.

16.1.4 Fluid Elements

16.1.4.1 Method of Derivation

The mathematical model for the compressible fluid consists of a gridwork of points at which pressures are calculated, and a set of "fluid elements" which are connected to the grid points and which fill the space occupied by the fluid. The model is analogous to a finite element structural model. The pressure is analogous to displacement and the pressure gradient is analogous to strain. The problem is assumed to have axial symmetry so that the "structural" model is defined by a set of fluid grid points in a plane that includes the axis of symmetry. Each fluid element occupies the interior of a circular ring that is concentric with the axis of symmetry. The surfaces of the ring are cones which intersect along circles that pass through the fluid grid points. For this reason the fluid grid points will also be referred to as fluid circles.

Due to the assumption of axial symmetry, the pressure within a fluid element can be expanded in a Fourier series with respect to the azimuth coordinate, ϕ,

$$p(r,\phi,z) = P^o + \sum_{n=1}^{N} P^n \cos n\phi + \sum_{n=1}^{N} P^{n*} \sin n\phi \quad . \tag{1}$$

The coefficients P^o, P^n and P^{n*} are functions of position in a radial plane. They will be expanded in truncated power series of the radial and axial coordinates. Thus,

$$P^n(r,z) = q_0^n + q_1^n r + q_2^n z + \dots \qquad n \geq 0 \quad , \tag{2}$$

$$P^{n*}(r,z) = q_0^{n*} + q_1^{n*} r + q_2^{n*} z + \dots \quad n \geq 1 \quad . \tag{3}$$

The coefficients q_j^n and q_j^{n*} are called the generalized coordinates of the element. They are selected to make the values of P^n and P^{n*} match the values of the pressure coefficients at the fluid grid points located at the corners of the element.

Thus, if P_i^n is the value of P^n at the i^{th} fluid grid point,

$$\{q_j^n\} = [H_{qp}^n] \{P_i^n\} \quad , \tag{4}$$

$$\{q_j^{n*}\} = [H_{qp}^n] \{P_i^{n*}\} \quad , \tag{5}$$

ORIGINAL PAGE IS OF POOR QUALITY

where $[H^n_{qp}]$ is a matrix of constant coefficients. Note that the matrices for the starred and unstarred coefficients are identical. Since $[H^n_{qp}]$ is evaluated from its inverse, see Section 16.1.4.2, it must be a square matrix, and the number of generalized coordinates for harmonic coefficient must be equal to the number of fluid grid points to which the element is attached.

The generalized potential energy contained within the volume of the element is, from Equation 15, Section 16.1.2,

$$U = \frac{1}{2} \int_V \frac{1}{\rho} (\nabla p \cdot \nabla p) \, dV \quad , \tag{6}$$

where the gradient of the pressure, ∇p, is the vector:

$$\nabla p = \frac{\partial p}{\partial r} \vec{e}_r + \frac{1}{r} \frac{\partial p}{\partial \phi} \vec{e}_\phi + \frac{\partial p}{\partial z} \vec{e}_z \quad . \tag{7}$$

Using Equation (1), the gradient is:

$$\begin{aligned} \nabla p = & \left[\frac{\partial P^o}{\partial r} + \sum_{n=1}^{N} \left(\frac{\partial P^n}{\partial r} \cos n\phi + \frac{\partial P^{n*}}{\partial r} \sin n\phi \right) \right] \vec{e}_r \quad , \\ & + \left[\sum_{n=1}^{N} \frac{n}{r} (-P^n \sin n\phi + P^{n*} \cos n\phi) \right] \vec{e}_\phi \quad , \\ & + \left[\frac{\partial P^o}{\partial z} + \sum_{n=1}^{N} \left(\frac{\partial P^n}{\partial z} \cos n\phi + \frac{\partial P^{n*}}{\partial z} \sin n\phi \right) \right] \vec{e}_z \quad . \end{aligned} \tag{8}$$

The integral over the volume consists of separate integrals over the angle, ϕ, and over the cross-sectional area, A. Thus,

$$U = \frac{1}{2} \int_A \left[\int_0^{2\pi} \frac{1}{\rho} \nabla p \cdot \nabla p \, r \, d\phi \right] dA \quad , \tag{9}$$

where $dA = dr\,dz$.

The inner integration results in zero values for all products of different harmonics and all sine-cosine products. The potential energy expression after applying the inner integration is:

906

$$U = \int \frac{\pi r}{\rho}\left[\left(\frac{\partial P^o}{\partial r}\right)^2 + \left(\frac{\partial P^o}{\partial z}\right)^2\right] dA$$

$$+ \sum_{n=1}^{N} \int \frac{\pi r}{2\rho}\left[\left(\frac{\partial P^n}{\partial r}\right)^2 + \left(\frac{\partial P^n}{\partial z}\right)^2 + \frac{n^2}{r^2}(P^n)^2\right] dA$$

$$+ \sum_{n=1}^{N} \int \frac{\pi r}{2\rho}\left[\left(\frac{\partial P^{n*}}{\partial r}\right)^2 + \left(\frac{\partial P^{n*}}{\partial z}\right)^2 + \frac{n^2}{r^2}(P^{n*})^2\right] dA \quad . \qquad (10)$$

Note that the harmonics, n, are uncoupled, and that the starred and unstarred terms are uncoupled.

The pressure coefficients P^o, P^n and P^{n*}, which vary throughout the element, are evaluated in terms of the constant coefficients (or generalized coordinates) q_j^o, q_j^n and q_j^{n*}. Thus the energy can be expressed as a quadratic function of these coordinates. The elements of the generalized stiffness matrix referred to the generalized coordinates for the nth harmonic order are

$$K_{ij}^{nq} = \frac{\partial^2 U}{\partial q_i^n \, \partial q_j^n} \quad . \qquad (11)$$

Using Equation 4, which expresses a linear relationship between the pressure coefficients at grid points and the generalized coordinates, the stiffness matrix for each harmonic is:

$$[K_{ij}^n] = [H_{qp}^n]^T \, [K_{ij}^{nq}] \, [H_{qp}^n] \quad . \qquad (12)$$

The stiffness matrix for the starred pressure coefficients is identically the same.

The equivalent mass matrix is derived in a similar manner. The "kinetic" energy, given by Equation 16 of Section 16.1.2, is first expanded in terms of the harmonic pressure coefficients. When the integration with respect to ϕ is carried out, the result is

$$T = T_o + \sum_{n=1}^{N} T_n + \sum_{n=1}^{N} T_n^* \quad , \qquad (13)$$

where

$$\left.\begin{aligned} T_o &= \int_A \frac{\pi r}{B} (\dot{p}^o)^2 \, dA \quad , \\ T_n &= \int_A \frac{\pi r}{2B} (\dot{p}^n)^2 \, dA \quad , \\ T_n^* &= \int_A \frac{\pi r}{2B} (\dot{p}^{n*})^2 \, dA \quad . \end{aligned}\right\} \qquad (14)$$

The pressure coefficients at interior points can, by combining Equation 2 with Equation 4 and Equation 3 with Equation 5, be expressed in terms of the pressure coefficients at the fluid grid points. The result is

$$\left.\begin{aligned} \dot{p}^n &= \sum_j f_j^n (r,z) \, \dot{p}_j^n \qquad n \geq 0 \quad , \\ \dot{p}^{n*} &= \sum_j f_j^n (r,z) \, \dot{p}_j^{n*} \qquad n \geq 1 \quad , \end{aligned}\right\} \qquad (15)$$

where $f_j^n (r,z)$ is an, as yet, undetermined function.

The terms in the mass matrix, referred to pressure coefficients at the fluid grid points, are

$$M_{ij}^n = \frac{\partial^2 T_n}{\partial \dot{p}_i^n \partial \dot{p}_j^n} = \frac{\pi}{B} \int_A f_i^n \, f_j^n \, r \, dA \qquad n \geq 1 \quad . \qquad (16)$$

The result for n = 0 is twice as large. The mass matrices for the starred terms are identical to those for the unstarred terms.

16.1.4.2 Triangular Fluid Element

A triangular fluid element contains the volume of fluid interior to a ring of triangular cross-section. Each circular edge of the ring defines a fluid point in the cross-section. The pressure in the element is assumed to be a linear function over the triangular section. A Fourier series is used to represent the variation of pressure around the circumference. The pressure is:

$$p(r,\phi,z) = p^o + \sum_{n=1}^{N} p^n \cos n\phi + \sum_{n=1}^{N} p^{n*} \sin n\phi \quad , \tag{17}$$

where the assumed pressure distribution over the cross section is:

$$\left.\begin{aligned} p^n(r,z) &= q_0^n + q_1^n r + q_2^n z \quad , \quad n \geq 0 \, , \\ p^{n*}(r,z) &= q_0^{n*} + q_1^{n*} r + q_2^{n*} z \quad , \quad n \geq 1 \, , \end{aligned}\right\} \tag{18}$$

where q_0^n, q_1^n, and q_2^n, $n = 0,1,\ldots,N$, and the corresponding starred terms, are the generalized coordinates of the element.

The transformation matrix $[H_{qp}^n]$, defined in Equation 4, is

$$[H_{qp}^n] = \begin{bmatrix} 1 & r_1 & z_1 \\ 1 & r_2 & z_2 \\ 1 & r_3 & z_3 \end{bmatrix}^{-1} \quad . \tag{19}$$

Substituting Equation 18 into Equation 10, the potential energy for each harmonic is:

$$\left.\begin{aligned} U_o &= \int_A \frac{\pi r}{\rho} \left[(q_1^o)^2 + (q_2^o)^2 \right] dA \quad , \\ U_n &= \int_A \frac{\pi r}{2\rho} \left[(q_1^n)^2 + (q_2^n)^2 + \frac{n^2}{r^2} (q_0^n + q_1^n r + q_2^n z)^2 \right] dA, \quad n > 0 \quad , \\ U_n^* &= \int_A \frac{\pi r}{2\rho} \left[(q_1^{n*})^2 + (q_2^{n*})^2 + \frac{n^2}{r^2} (q_0^{n*} + q_1^{n*} r + q_2^{n*} z)^2 \right] dA, \quad n > 0 \quad . \end{aligned}\right\} \tag{20}$$

The terms in the upper half of the symmetrical stiffness matrix referred to the generalized coordinates are, from Equations 11 and 20,

$$\left.\begin{aligned}
&n = 0: \\
&\qquad K_{11}^{0q} = K_{22}^{0q} = \frac{2\pi}{\rho} I_{20} \quad , \\
&\qquad K_{00}^{0q} = K_{01}^{0q} = K_{02}^{0q} = K_{12}^{0q} = 0 \quad ,
\end{aligned}\right\} \qquad (21)$$

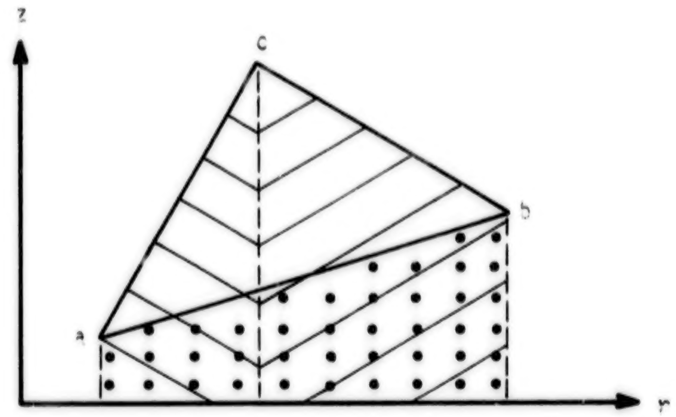
$$\begin{aligned}
&n > 0:
\end{aligned}$$

$$\left.\begin{aligned}
K_{00}^{nq} &= \frac{n^2\pi}{\rho} I_{00} \quad , \\
K_{01}^{nq} &= \frac{n^2\pi}{\rho} I_{10} \quad , \\
K_{02}^{nq} &= \frac{n^2\pi}{\rho} I_{01} \quad , \\
K_{11}^{nq} &= \frac{\pi}{\rho} (1 + n^2) I_{20} \quad , \\
K_{12}^{nq} &= \frac{n^2\pi}{\rho} I_{11} \quad , \\
K_{22}^{nq} &= \frac{\pi}{\rho} (I_{20} + n^2 I_{02}) \quad ,
\end{aligned}\right\} \qquad (22)$$

and similarly for the starred terms. The $I_{k\ell}$ coefficients are the following integrals over the cross section:

$$I_{k\ell} = \int_A r^{k-1} z^{\ell} \, dA, \qquad k \geq 0 \quad , \quad \ell \geq 0 \quad . \qquad (23)$$

The integrals $I_{k\ell}$ may be evaluated by dividing the triangular surface into three trapezoids as shown below:



The three points a, b and c are described by their r and z coordinates at $\phi = 0$. The integrals take the form:

$$I_{k\ell} = G_{k\ell}(a,b) + G_{k\ell}(b,c) + G_{k\ell}(c,a) \quad . \tag{24}$$

The general trapezoidal integral $G_{k\ell}$ is derived as follows. Let $G_{k\ell}$ be evaluated for two points $p_1 = r_1, z_1$ and $p_2 = r_2, z_2$. Then

$$G_{k\ell} = \int_{r_2}^{r_1} \left[\int_{z=0}^{z=z_1 + \frac{z_2-z_1}{r_2-r_1}(r-r_1)} r^{k-1} z^{\ell} \, dz \right] dr \quad . \tag{25}$$

Performing the integration with respect to z results in the expression:

$$G_{k\ell} = \frac{1}{\ell+1} \int_{r_2}^{r_1} r^{k-1} \left[z_1 + \frac{z_2-z_1}{r_2-r_1}(r - r_1) \right]^{\ell+1} dr \quad . \tag{26}$$

A convenient set of parameters is:

$$\left.\begin{aligned} \Delta r &= r_2 - r_1 \quad , \\ \Delta z &= z_2 - z_1 \quad , \\ \beta &= z_1 \frac{\Delta r}{\Delta z} - r_1 \quad . \end{aligned}\right\} \tag{27}$$

Rearranging the integrand of Equation 26 in terms of these parameters

$$G_{k\ell} = \frac{1}{\ell+1}\left(\frac{\Delta z}{\Delta r}\right)^{\ell+1} \int_{r_2}^{r_1} (\beta + r)^{\ell+1}\, r^{k-1}\, dr \quad . \tag{28}$$

The polynomial expansion of a binomial raised to a power is a series of the form:

$$(\beta + r)^{\ell+1} = \sum_{j=0}^{\ell+1} \frac{(\ell+1)!\, \beta^{\ell+1-j}\, r^j}{(j!)(\ell+1-j)!} \quad . \tag{29}$$

Substituting Equation 28 into Equation 27 and rearranging gives a series of simple integrals:

$$G_{k\ell} = \frac{1}{\ell+1}\left(\frac{\Delta z}{\Delta r}\right)^{\ell+1} \sum_{j=0}^{\ell+1}\left(\frac{(\ell+1)!\, \beta^{\ell+1-j}}{j!\ (\ell+1-j)!} \int_{r_2}^{r_1} r^{k-1+j}\, dr\right) = \frac{1}{\ell+1}\left(\frac{\Delta z}{\Delta r}\right)^{\ell+1} \sum_{j=0}^{\ell+1} C_j \quad . \tag{30}$$

The term for j = 0 requires special treatment. Thus,

$$C_o = \beta^{\ell+1} \int_{r_2}^{r_1} r^{-1}\, dr = \beta^{\ell+1} \log \frac{r_1}{r_2} \quad , \quad k = 0 \quad ,$$

and

$$C_o = \beta^{\ell+1} \int_{r_2}^{r_1} r^{k-1}\, dr = \frac{\beta^{\ell+1}}{k}\left(r_1^k - r_2^k\right) , \ k \neq 0 \quad . \tag{31}$$

The complete series is

$$G_{k\ell} = \frac{1}{\ell+1}\left(\frac{\Delta z}{\Delta r}\right)^{\ell+1}\left[C_o + \sum_{j=1}^{\ell+1}\frac{(\ell+1)!\,\beta^{\ell+1-j}\left(r_1^{k+j} - r_2^{k+j}\right)}{j!\,(\ell+1-j)!\,(k+j)}\right] \quad . \qquad (32)$$

If $\Delta r = 0$, $G_{k\ell}$ should be set equal to zero.

In the evaluation of the mass matrix by means of Equation 16, the function f_1^n may be written

$$f_1^n = \frac{r(z_2 - z_3) + r_2(z_3 - z) + r_3(z - z_2)}{r_1(z_2 - z_3) + r_2(z_3 - z_1) + r_3(z_1 - z_2)} \quad , \qquad (33)$$

which is a linear function of r and z that equals unity at (r_1, z_1) and vanishes at (r_2, z_2) and (r_3, z_3). f_2^n and f_3^n are similar with appropriate permutations of subscripts. The final result, for both starred and unstarred terms, is

$$M_{ij}^n = \frac{\pi A}{60B}(r_1 + r_2 + r_3 + r_i + r_j)(1 + \delta_j^i) \quad , \quad n > 0 \quad , \qquad (34)$$

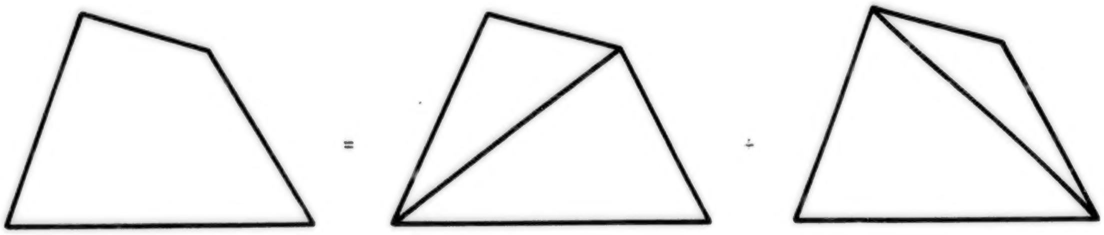
where

$$\delta_j^i = 1, \quad \text{if } i = j$$
$$= 0, \quad \text{if } i \neq j$$

and A is the surface area in the (r,z) plane. The mass matrix for n = 0 is just twice as large.

16.1.4.3 Quadrilateral Fluid Element

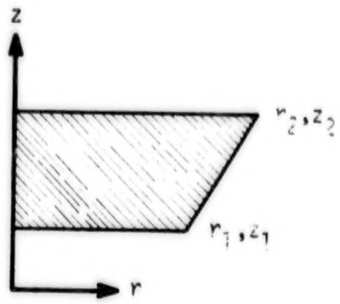
The quadrilateral fluid element consists of four triangular fluid elements defined by the exterior edges and diagonals of the quadrilateral as shown below:



The "stiffness" and "mass" matrices for the quadrilateral are taken as one-half of the sum of stiffness and mass matrices for the individual triangles.

16.1.4.4 Center Element

Triangular and quadrilateral fluid elements are not permitted to be connected to a fluid point on the axis of symmetry because the resulting pressures will be inconsistent and the element integrals may be singular. For safety in computation it is not recommended that a fluid grid point be placed close to the axis. Instead a special element may be used which is bounded by two parallel planes perpendicular to the axis and a conical outer boundary. The geometry of the element is specified by two rings at the outer edge of the upper and lower planes as shown:



The pressure is assumed to have the following functional form, which is an asymptotic solution of the wave equation as r → 0:

$$\begin{aligned} p(r,\phi,z) &= P^0 + \sum_{n=1}^{N} P^n \cos n\phi + \sum_{n=1}^{N} P^{n*} \sin n\phi \\ &= (q_1^0 + q_2^0 z) + \sum_{n=1}^{N} (r)^n (q_1^n + q_2^n z) \cos n\phi \\ &\quad + \sum_{n=1}^{N} (r)^n (q_1^{n*} + q_2^{n*} z) \sin n\phi \quad . \end{aligned} \tag{35}$$

This function disappears at the axis for all harmonics except zero. Its gradient approaches zero at the axis for all harmonics except the zero and first harmonics.

The relation between the generalized coordinates q_j^n and the harmonic coefficients of the pressure on the two corner rings is:

$$\begin{Bmatrix} q_1^n \\ q_2^n \end{Bmatrix} = \begin{bmatrix} (r_1)^n & (r_1)^n z_1 \\ (r_2)^n & (r_2)^n z_2 \end{bmatrix}^{-1} \begin{Bmatrix} P_1^n \\ P_2^n \end{Bmatrix} \tag{36}$$

The generalized potential energy for each harmonic, according to Equation 10, is:

$$\left.\begin{aligned} U_o &= \int_A \frac{\pi r}{\rho}\, q_2^0{}^2\, dA \quad , \\ U_n &= \int_A \frac{\pi}{2\rho}\left[2n^2\, r^{(2n-1)} \left(q_1^n + q_2^n z\right)^2 + r^{(2n+1)} \left(q_2^n\right)^2\right] dA \qquad n > 0 \, , \\ U_n^* &= \int_A \frac{\pi}{2\rho}\left[2n^2\, r^{(2n-1)} \left(q_1^{n*} + q_2^{n*} z\right)^2 + r^{(2+1)} \left(q_2^{n*}\right)^2\right] dA \qquad n > 0 \, . \end{aligned}\right\} \tag{37}$$

The stiffness matrix terms, K_{ij}^{qn}, extracted by using Equation 11, are:

$$\left.\begin{aligned} K_{11}^{qn} &= \frac{2\pi}{\rho}\, n^2\, I_{2n,0} \quad , \\ K_{12}^{qn} &= K_{21}^{qn} = \frac{2\pi}{\rho}\, n^2\, I_{2n,1} \quad , \\ K_{22}^{qn} &= \frac{\pi}{\rho}\left(2n^2\, I_{2n,2} + I_{2n+2,0}\right) , \end{aligned}\right\} \quad n > 0 \tag{38}$$

where $I_{k\ell}$ is defined in Equation 23. The starred terms are identical to the above and the zero harmonic ($n = 0$) terms are multiplied by 2.

The integrals $I_{k\ell}$ are evaluated as follows. From the definition of the integral and the shape of the region

$$I_{k\ell} = \int_{z_1}^{z_2} \left[\int_{r=0}^{r = r_1 + \left(\frac{r_2 - r_1}{z_2 - z_1}\right)(z - z_1)} r^{k-1}\, z^{\ell}\, dr \right] dz \tag{39}$$

After performing the integration with respect to r,

$$I_{k\ell} = \frac{1}{k} \int_{z_1}^{z_2} \left(r_1 + \frac{r_2 - r_1}{z_2 - z_1} (z - z_1) \right)^k z^\ell \, dz \quad , \qquad k > 0 \quad . \tag{40}$$

From Equation 38 it is seen that the integrals for k = 0 are not actually required since their coefficients are zero. Define $\Delta r = r_2 - r_1$ and $\Delta z = z_2 - z_1$, and integrate Equation 40 by parts with

$$\begin{aligned} du &= \left(r_1 + \frac{\Delta r}{\Delta z} (z - z_1) \right)^k dz \quad , \\ v &= z^\ell \quad . \end{aligned} \tag{41}$$

The result is

$$I_{k\ell} = \frac{\Delta z}{k(k+1)\Delta r} \left[\left(r + \frac{\Delta r}{\Delta z} (z - z_1) \right)^{k+1} z^\ell \Bigg|_{z_1}^{z_2} - \int_{z_1}^{z_2} \left(r_1 + \frac{\Delta r}{\Delta z} (z - z_1) \right)^{k+1} z^{\ell-1} \, dz \right] . \tag{42}$$

The integral term in Equation 42 is replaced by zero for $\ell = 0$.

The integration by parts is performed ℓ times, i.e., until the exponent of z in the integral is zero. The result may then be expressed as

$$\begin{aligned} I_{k\ell} = \frac{\Delta z}{k(k+1)\Delta r} \Bigg[\left(r_2^{k+1} z_2^\ell - r_1^{k+1} z_1^\ell \right) - \frac{\Delta z}{(k+2)\Delta r} \Bigg[\left(r_2^{k+2} z_2^{\ell-1} - r_1^{k+2} z_1^{\ell-1} \right) \\ - \frac{\Delta z}{(k+3)\Delta r} \Bigg[\text{-------} - \frac{\Delta z}{(k+\ell+1)\Delta r} \left[r_2^{k+\ell+1} - r_1^{k+\ell+1} \right] \text{----} \Bigg] \Bigg] \Bigg] \quad . \end{aligned} \tag{43}$$

Equation 43 has $\ell+1$ closing brackets. The following approximation may be used for the special case in which $\Delta r / r_1 << 1$.

$$I_{k\ell} = \int_{z_1}^{z_2} r_1^k z^\ell \, dz = \frac{r_1^k \left(z_2^{\ell+1} - z_1^{\ell+1} \right)}{k(\ell+1)} \quad . \tag{44}$$

The stiffness matrix referred to the harmonic coefficients of pressure at the fluid grid points is

$$[K^n_{ij}] = [H^n_{qp}]^T \, [K^{qn}_{ij}] \, [H^n_{qp}] \quad . \tag{45}$$

The elements of $[H^n_{qp}]^{-1}$ are shown in Equation 36.

The mass matrix associated with the center element is generated from the kinetic energy integrals, Equation 14. Substituting

$$\dot{p}^n = (r)^n \left(\dot{q}^n_1 + \dot{q}^n_2 \, z\right) \quad , \tag{46}$$

into Equation 14, the generalized kinetic "energy" of the element for each harmonic is

$$T_n = \int \frac{\pi r}{2B} (r)^{2n} \left(\dot{q}^n_1 + \dot{q}^n_2 \, z\right)^2 dA \quad , \quad n > 0 \quad . \tag{47}$$

The mass matrix terms for the generalized coordinates are calculated from the equation:

$$M^{qn}_{ij} = \frac{\partial^2 \, T_n}{\partial \dot{q}^n_i \, \partial \dot{q}^n_j} \quad . \tag{48}$$

In terms of the previously defined integral parameters, $I_{k\ell}$, the terms in the mass matrix for the generalized coordinates are:

$$\left.\begin{aligned} M^{qn}_{11} &= \frac{\pi}{B} I_{2n+2,0} \quad , \\ M^{qn}_{12} &= M^{qn}_{21} = \frac{\pi}{B} I_{2n+2,1} \quad , \\ M^{qn}_{22} &= \frac{\pi}{B} I_{2n+2,2} \quad . \end{aligned}\right\} \quad n > 0 \tag{49}$$

The elements of the starred mass matrix are identical. For n = 0 the above results are multiplied by 2. The mass matrix referred to the harmonic coefficients of pressure at the fluid grid points is

$$[M^n_{ij}] = [H^n_{qp}]^T \, [K^{nq}_{ij}] \, [H^n_{qp}] \quad . \tag{50}$$

16.1.5 Boundary Matrices

The relation between the pressures, $\{p_j\}$, forces, $\{F_i\}$, and displacements, $\{u_i\}$, at the structural boundary, given for an individual point by Equation 15 of Section 16.1.3, has the general matrix form

$$\{F_i\} = [A]\{p_j\} - [K_g]\{u_i\} \quad . \tag{1}$$

The terms in the vector $\{F_i\}$ are the forces on the displacements, $\{u_i\}$. The $[K_g]$ matrix is a function of the gravity.

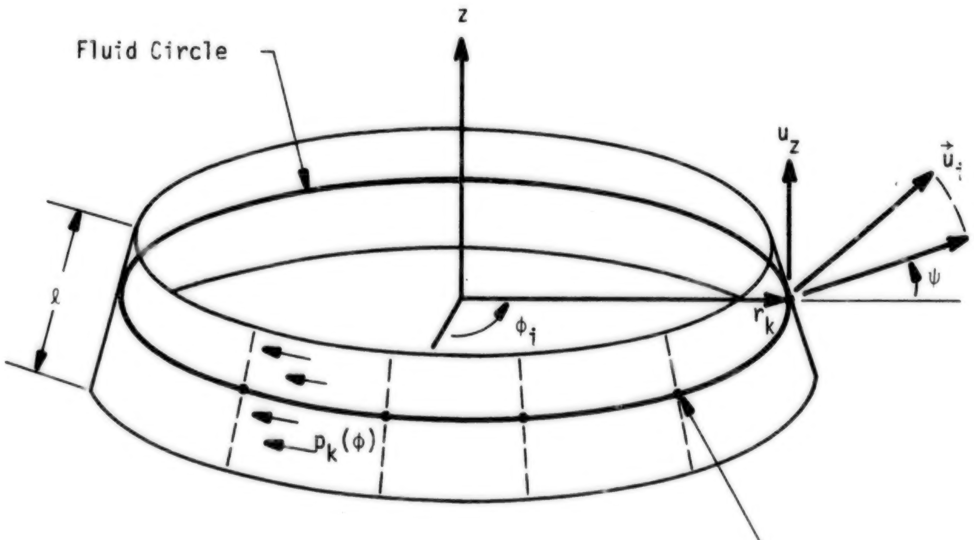
The generalized forces on the fluid, $\{I_j\}$, corresponding to pressures, $\{p_j\}$, are dependent on the structural acceleration, $\{\ddot{u}_i\}$, as shown by Equation 7 of Section 16.1.3. The general matrix relation has the form

$$\{I_j\} = s^2 \, [R]\{u_i\} \quad , \tag{2}$$

where s = d/dt. In order to conserve the energy flow, $\{u_i\}^T\{p_i\}$, through the surface, it is necessary that

$$[R] = -[A]^T \quad . \tag{3}$$

Several structural points are attached to each boundary fluid circle as shown in the following figure.



where r_k is the radius of the circle

ℓ is the slant length of an equivalent conical surface associated with the circle

ϕ_i is the angular position of point i

$\vec{u}_i$ is the displacement vector of point i

$p_k(\phi)$ is the pressure around the k^{th} fluid circle

$\vec{n}$ is the vector normal to the equivalent conical surface

ψ is the angle between $\vec{n}$ and the (r,ϕ) plane

The pressure function around the circle is

$$p_k(\phi) = P_k^o + \sum_n \left(P_k^n \cos n\phi + P_k^{n*} \sin n\phi \right) . \tag{4}$$

The outward normal force on an increment of area, given by Equation 15, Section 16.1.3, is

$$dF_i = [p(\phi) - \rho\vec{g}\cdot\vec{u}(\phi)]\, dS_i \quad . \tag{5}$$

Thus the total outward force acting on the element of surface area associated with the i^{th} structural grid point can be found by integrating

$$F_i = \int_{S_i} dF_i = \int_{S_i} \left[P_k^o + \sum_n P_k^n \cos n\phi + P_k^{n*} \sin n\phi - \rho g u_z \right] dS_i \quad . \tag{6}$$

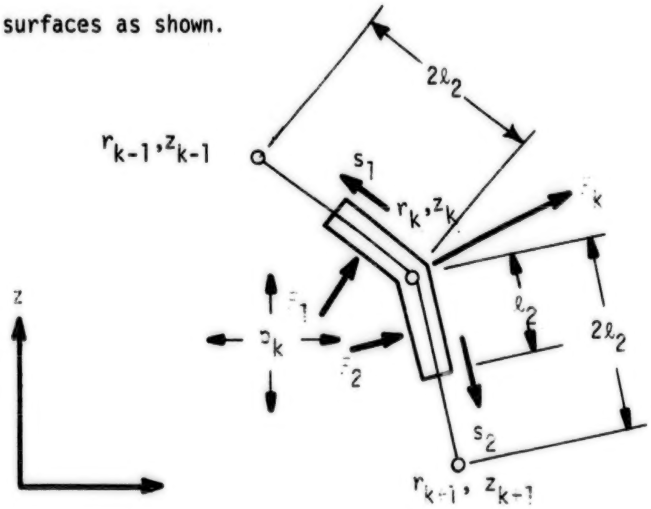
The vertical displacement u_z can be expressed in terms of the available components of structural displacement, u_j. Thus

$$F_i = A_i^o P_k^o + \sum_n \left(A_i^n P_k^n + A_i^{n*} P_k^{n*} \right) - \sum_j K_{ij} u_j \quad , \tag{7}$$

where the coefficients A_i^o, A_i^n, A_i^{n*} and K_{ij} will be found by expanding Equation 6 and comparing the result with Equation 7.

In order to produce the coefficients in Equation 7, we must define the limits of integration in Equation 6 and the manner in which the direction of the normal varies in the vicinity of the fluid circle. The pressure function of a fluid circle is assumed to be a function of ϕ only and acts over the area as defined below.

The boundary is assumed to consist of conical surfaces interconnecting fluid circles as shown below. The forces acting on a fluid circle are taken as the forces acting on the adjacent halves of the adjacent conical surfaces as shown.



The forces on each of the two sections, for a unit increment of azimuth angle, are

$$\left.\begin{aligned} F_1 &= \int_0^{\ell_1} p\, r_1\, ds_1 \quad , \\ F_2 &= \int_0^{\ell_2} p\, r_2\, ds_2 \quad , \end{aligned}\right\} \tag{9}$$

where

$$\left.\begin{aligned} \ell_1 &= \frac{1}{2}\sqrt{(z_{k-1} - z_k)^2 + (r_{k-1} - r_k)^2} \ , \\ \ell_2 &= \frac{1}{2}\sqrt{(z_k - z_{k+1})^2 + (r_k - r_{k+1})^2} \ , \\ r_1 &= r_k - s_1 \frac{(r_k - r_{k-1})}{2\ell_1} \quad , \\ r_2 &= r_k + s_2 \frac{(r_{k+1} - r_k)}{2\ell_2} \quad . \end{aligned}\right\} \tag{10}$$

Performing the integrations indicated in Equation 9 we obtain:

$$F_1 = p\ell_1\, r_k \left[1 - \frac{1}{4}\left(\frac{r_k - r_{k-1}}{r_k}\right)\right] \quad ,$$

$$F_2 = p\ell_2\, r_k \left[1 + \frac{1}{4}\left(\frac{r_{k+1} - r_k}{r_k}\right)\right] \quad . \tag{11}$$

Resolving the forces into r and z components and adding them, produces the radial and vertical components of the force on the ring:

$$F_r = p\, r_k\, \Delta z \quad ,$$

$$F_z = p\, r_k\, \Delta r \quad , \tag{12}$$

where

$$\Delta z = \ell_1 \left[1 - \frac{1}{4}\left(\frac{r_k - r_{k-1}}{r_k}\right)\right]\left(\frac{z_{k-1} - z_k}{2\ell_1}\right) + \ell_2 \left[1 + \frac{1}{4}\left(\frac{r_{k+1} - r_k}{r_k}\right)\right]\left(\frac{z_k - z_{k+1}}{2\ell_2}\right) \; ,$$

$$\Delta r = \ell_1 \left[1 - \frac{1}{4}\left(\frac{r_k - r_{k-1}}{r_k}\right)\right]\left(\frac{r_k - r_{k-1}}{2\ell_1}\right) + \ell_2 \left[1 + \frac{1}{4}\left(\frac{r_{k+1} - r_k}{r_k}\right)\right]\left(\frac{r_{k+1} - r_k}{2\ell_2}\right) \; . \tag{13}$$

An equivalent conical section, which produces the same magnitude and direction of force, has a slant length, ℓ, and cone angle, ψ, given by:

$$\ell = \sqrt{\Delta r^2 + \Delta z^2} \quad ,$$

$$\psi = \text{Arctan}\left(\frac{\Delta r}{\Delta z}\right) \quad . \tag{14}$$

The structural grid points are, in general, placed at irregularly spaced azimuth angle, ϕ_i, around the fluid circle. The limits of integration for the i^{th} grid point are $\phi_{1i} = (\phi_{i+1}+\phi_i)/2$ and $\phi_{0i} = (\phi_i+\phi_{i-1})/2$. Thus, using the results expressed by Equation 14 in Equation 6, the force applied to the i^{th} grid point in the direction of the normal, $\vec{n}$, is

$$F_i = r_k \ell \int_{\phi_{0i}}^{\phi_{1i}} \left[P_k^o + P_k^n \cos n\phi + P_k^{n*} \sin n\phi - \rho g \left(u_{ni} \sin \psi + u_{ti} \cos \psi\right)\right] d\phi \quad , \tag{15}$$

where u_{ni} is the normal component of displacement and u_{ti} is the tangential component of displacement at the i^{th} grid point. Note that the assumption is made, implicitly, that the angle $(\phi - \phi_i)$ between the local normal and the radial direction at the i^{th} grid point is small enough that $\cos(\phi - \phi_i)$ may be assumed equal to unity.

Performing the indicated integration, the coefficients in Equation 7 are:

$$\left.\begin{aligned} A_i^o &= r_k \ell \,(\phi_{1i} - \phi_{0i}) \quad , \\ A_i^n &= \frac{r_k \ell}{n}\,(\sin \phi_{1i} - \sin \phi_{0i}) \quad , \\ A_i^{n*} &= \frac{r_k \ell}{n}\,(\cos \phi_{0i} - \cos \phi_{1i}) \quad . \end{aligned}\right\} \tag{16}$$

The components of force in a cylindrical (r, ϕ, z) coordinate system are

$$\begin{Bmatrix} F_{ir} \\ F_{i\phi} \\ F_{iz} \end{Bmatrix} = \begin{Bmatrix} \cos \psi \\ 0 \\ \sin \psi \end{Bmatrix} F_i \quad . \tag{17}$$

In order to produce loads on the point in any other coordinate system, the vector in Equation 17 is multiplied by a transformation matrix. The stiffness coefficient in Equation 7 that relates normal force, F_i, to motion in the direction of the normal is

$$K_{ii} = r_k \ell \rho g \sin \psi \,(\phi_{1i} - \phi_{0i}) \quad . \tag{18}$$

The off-diagonal term that relates normal force to tangential motion, u_t, is

$$K_{it} = r_k \ell \rho g \cos \psi \,(\phi_{1i} - \phi_{0i}) \quad . \tag{19}$$

These coefficients are rotated into the directions taken locally by the global coordinate system.

As an example, the boundary stiffness matrix in a cylindrical (r,ϕ,z) coordinate system is, for an individual structural grid point,

$$[K_{ij}] = r_k \ell \rho g \, (\phi_{1i} - \phi_{0i}) \begin{array}{c} \begin{array}{ccc} (r) & (\phi) & (z) \end{array} \\ \left[\begin{array}{ccc} 0 & 0 & \cos\psi \\ 0 & 0 & 0 \\ 0 & 0 & \sin\psi \end{array}\right] \end{array} \begin{array}{c} \\ (r) \\ (\phi) \\ (z) \end{array} \quad . \qquad (20)$$

In summary, the arrays which describe the interconnection of the fluid and the structure are the matrices of the coefficients A_i^o, A_i^n, $A_i^{n^*}$, and K_{ij} defined in Equation 7 and evaluated in Equations 16 and 20. The A_i^o, A_i^n, and $A_i^{n^*}$ coefficients also serve, by virtue of Equations 2 and 3, as the arrays that give the generalized forces on the harmonic pressure coefficients due to the structural displacements. In the latter capacity they have the form of a mass matrix. NASTRAN includes a separate functional module, the Boundary Matrix Generator, for the generation of these arrays. The outputs of the module are treated as direct input matrices by the Direct Dynamic Matrix Assembler (GKAD). See Section 9.3.3 for details.

The boundary condition at a free fluid surface is given by Equation 13 of Section 16.1.3. Neglecting the static term, it is seen that the boundary condition describes a surface "mass" distribution with surface density equal to $1/\rho g$. If the gravity field is absent $(g = 0)$, the pressure is zero at the surface. This condition is treated automatically in NASTRAN by applying single point constraints to the degrees of freedom that are the harmonic pressure coefficients at the free surface.

When gravity is present, the program satisfies the free surface boundary condition by connecting a mass matrix to the harmonic pressure coefficients at surface grid points. The elements in the mass matrix are obtained by an energy method. The pressure is assumed to vary linearly with radius between two adjacent fluid circles on the free surface, so that the pressure coefficient is

$$p^n = \frac{r - r_1}{r_2 - r_1} p_2^n + \frac{r_2 - r}{r_2 - r_1} p_1^n \quad , \qquad (21)$$

between the two fluid circles. The "kinetic" energy for the n^{th} harmonic coefficient is, by analogy with Equation 14 of Section 6.1.4

$$T_n = \frac{\pi}{2\rho g} \int_{r_1}^{r_2} (\dot{P}^n)^2 \, r \, dr \quad . \qquad (22)$$

The elements of the mass matrix interconnecting P_1^n and P_2^n are then obtained from

$$M_{ij} = \frac{\partial^2 T_n}{\partial \dot{P}_i^n \, \partial \dot{P}_j^n} \quad . \qquad (23)$$

The results are, for both the starred and unstarred coefficients,

$$\left.\begin{aligned} M_{11}^n &= \frac{\pi(r_2 - r_1)}{12\rho g} (3r_1 + r_2) \quad , \\ M_{12}^n = M_{21}^n &= \frac{\pi(r_2 - r_1)}{12\rho g} (r_1 + r_2) \quad , \\ M_{22}^n &= \frac{\pi(r_2 - r_1)}{12\rho g} (3r_2 + r_1) \quad . \end{aligned}\right\} \quad n > 0 \qquad (24)$$

The values are multiplied by two for $n = 0$.

For the region between the axis of symmetry and the first fluid circle, the pressure coefficients are assumed to be proportional to the n^{th} power of the radius (see Section 16.1.4.4). Thus, in this region,

$$P^n = P_1^n \left(\frac{r}{r_1}\right)^n \quad , \qquad (25)$$

and

$$T_n = \frac{\pi}{2\rho g} \left(\dot{P}_1^n\right)^2 \int_0^{r_1} r \left(\frac{r}{r_1}\right)^{2n} dr, \qquad n > 0 \quad . \qquad (26)$$

The element of the mass matrix is

$$M_{11}^n = \frac{\partial^2 T_n}{\left(\partial \dot{P}_1^n\right)^2} = \frac{\pi \, r_1^2}{(2n+2)\rho g} \quad , \qquad n > 0 \quad , \qquad (27)$$

while for n = 0

$$M^{o}_{11} = \frac{\pi r_1^2}{\rho g} \quad . \tag{28}$$

ORIGINAL PAGE IS
OF POOR QUALITY

16.1.6 Applications

The hydroelastic capability described in the preceding subsections may include many different effects such as variable density and compressibility, a free surface, rigid or flexible structural boundaries, multiple boundaries, free surfaces with gravity, and user defined matrices for the simulation of viscous effects and auxiliary connections to the fluid.

The general restrictions on the problem to be solved are: 1) the fluid must have axisymmetric geometry and properties; the motions, however, may be general; 2) the mathematics are valid only for small displacements; 3) the solution is accomplished only for dynamic pertubations from the static equilibrium. Both of the latter restrictions rule out direct solution of problems with large steady-state velocities. Indirect methods may be applied to these problems, but they must be provided by the user via direct input matrices.

In order to model fluid problems, a number of special considerations must be kept in mind:

1. The degrees of freedom for the fluid are harmonic pressure coefficients. The generalized forces acting on these degrees of freedom are actually the accelerations of the fluid. The applications of a dynamic force is not allowed except through a structural boundary.

2. At a free surface the gravity effect is modeled automatically with finite elements. If gravity is missing the pressures at a free surface will be constrained to zero. If gravity is present, the geometry of the free surface must be compatible with static equilibrium, i.e. it must lie in a plane perpendicular to the axis of symmetry.

3. The structural boundary may be either rigid or flexible. A fluid with no boundary definition in the input data will have a fixed rigid boundary.

4. The effects of variable fluid density and of a gravity field are not completely compatible. If the effects of either gravity or variable density are small they may be used together but second order errors may result.

The hydroelastic capability may be used with any of the available rigid formats. The rigid formats for static analysis and for elastic stability analysis will, however, produce trivial results, since the fluid-structure interaction matrices are ignored by these rigid formats. Recommended Rigid Formats and the restrictions on each are described below:

Rigid Format No. 3 - Normal Modes Analysis

The modes of a fluid in a rigid container may be extracted with this rigid format. Any structural data in the deck will be treated as a disjoint problem. (The structure may also produce normal modes.) Free surface effects with or without gravity will be accounted for.

Rigid Format No. 7 - Direct Complex Eigenvalue Analysis

The combined modes of the fluid and structure are obtained with this rigid format. If no damping or direct input matrices are added, the resulting complex roots will be purely imaginary numbers, whose values are the natural frequencies of the system. The mode shape may be normalized to maximum quantity (pressure or displacement) or to a specified structural displacement.

Rigid Format No. 8 - Direct Frequency and Random Response

It should be remembered that loads may be applied only to structural grid points. The use of overall structural damping (parameter g) is not recommended since the fluid matrices will be affected.

Rigid Format No. 9 - Direct Transient Response

Transient analysis may be performed for the fluid-structure system. The following rules apply.

1. Applied loads and initial conditions may only be given for the structural degrees of freedom.

2. All quantities are measured relative to static equilibrium. The initial values of the pressures are assumed to be in equilibrium.

3. Overall structural damping (parameters ω_3 and g) should not be used.

Rigid Formats 10, 11, and 12 - Modal Formulations

Although these rigid formats may be used in a fluid-structure interaction problem, their practicality is limited. The modal coordinates used to formulate the dynamic matrices are the normal modes of both the fluid and the structure solved as uncoupled systems. Even though the range of natural frequencies is typically very different for the fluid and the structure, a similar number of modes should be chosen from each. The safest method with the present version of NASTRAN is the extraction of a large number of modes using the Tridiagonal Method. This procedure, however, results in a dynamic system with large full matrices. The Direct Formulation is probably more efficient.

In applying the hydroelastic capability, the user selects the harmonic coefficients for the representation of the fluid. There is no requirement relating the number of harmonics and the number of structural grid points around the circumference nor is it required that the structural grid points be evenly spaced. If Ng is the number of structural grid points around a circumference, then the results for harmonic orders greater than $\frac{1}{2}$ (Ng - 1) will, however, be not meaningful for that circumference since these harmonics are not excited. For problems with transverse symmetry planes it will frequently be possible to eliminate the starred (sine) harmonic coefficients, i.e., $\frac{\partial P}{\partial \phi} = 0$ at $\phi = 0$ or π.

The program includes a provision whereby advantage may be taken of one or more planes of structural symmetry. The user specifies an even number m, where 1/m is the fraction of the circumference used in the structural model. The boundary conditions at the radial planes bounding the structural model may be both symmetric, both antisymmetric or one of each. The matrix, $[A]^T$, defining the generalized forces on the pressure coefficients is multiplied by m. The matrices defining the fluid forces on the structure are unchanged. It is required that the pressure harmonics selected by the user be consistent with the symmetry assumptions. Thus, if m = 6 and symmetric boundary conditions are placed at $\phi = 0$ and $\phi = \pi/3$, the permissible values of n are 0, 3, 6, 9, etc.

At present the axisymmetric structural elements (conical shell, doubly curved shell, and solid of revolution) cannot be used in hydroelastic problems. All other structural elements are available.

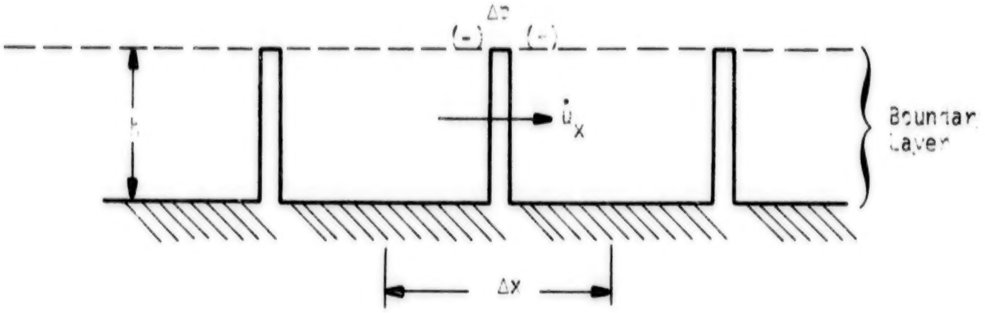
The topology of the fluid surface is quite arbitrary. Figure 1 shows several examples of permissible topologies. The major restriction is that, when the problem contains two or more tanks, they must have the same axis of symmetry. As an aid in establishing topology, the user must provide separate sequential lists of free surface points and structural boundary points for each disjoint boundary segment. The fluid must be located to the right of the boundary when the boundary is traversed according to the sequence in the list. Rigid inclusions or rigid external boundary segments, are not included in the lists. The existence of a rigid boundary is implied simply by the absence of elements in the boundary matrices.

The input data consists of card entries that describe the following.

1. The cross-sectional coordinates of fluid grid points (fluid circles).
2. Connection information and physical properties (density and bulk modulus) for fluid elements.
3. Identification of structural grid points.
4. Sequential lists of fluid circles on free surfaces and on structural boundaries.
5. The harmonic components of pressure to be considered.
6. The magnitude of the gravity vector.
7. Identification of radial planes of symmetry.
8. The location (radius and azimuth) of points on free surfaces at which normal displacements are desired for output.
9. The location of points in the fluid at which values of the pressure are desired for output (pressure points).
10. Direct input matrix terms coupling fluid degrees of freedom with each other or with any other structural degree of freedom or extra points.

A useful special feature of the program is that the specification of a pressure point in the fluid, item 9 above, may be used in conjunction with direct matrix input, item 10, to define a pressure transducer in a servo control system. In this application the pressure point is related to the harmonic pressure coefficients by the analyst using a multipoint equation of constraint.

Another use of direct matrix input occurs in the treatment of viscous damping due to tank wall friction. The effect may be represented by viscous dampers interconnecting the harmonic pressure coefficients in the fluid, and the resulting matrix of damping coefficients may be inserted by means of the direct matrix input feature. The arrangement of the viscous dampers is derived as follows. Let the frictional effect be represented by uniformly spaced porous baffle plates normal to the wall as shown below.



The frictional force on the wall per unit of surface area is,

$$f = - h \frac{\Delta p}{\Delta x} = C \dot{u}_x \quad . \tag{1}$$

The height of the baffle, h, is made equal to the thickness of the boundary layer. Considering the sides of the baffle to be external fluid surfaces, Equation 7 of Section 16.1.3 may be used to calculate the generalized force transferred through the baffle,

$$I_x = - S \ddot{u}_x = - S s \dot{u}_x \quad , \tag{2}$$

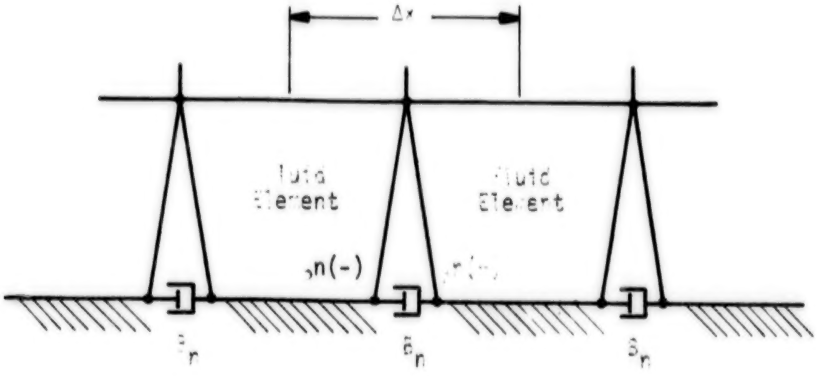
where S is the area of the baffle and s = d/dt. Substituting for $\dot{u}_x$ from Equation 1,

$$I_x = \frac{Sh}{C\Delta x} \Delta(\dot{p}) \tag{3}$$

Substituting harmonic coefficients for I_x and $\dot{p}$, and integrating over the circumference,

$$\left. \begin{aligned} I_x^n &= \frac{\pi r h^2}{C\Delta x} \Delta(\dot{p}^n) \qquad n > 0 \quad , \\ I_x^o &= \frac{2\pi r h^2}{C\Delta x} \Delta(\dot{p}^o) \qquad n = 0 \quad . \end{aligned} \right\} \tag{4}$$

The coefficient $B_n = \frac{\pi r h^2}{C\Delta x} (1+\delta_o^n)$ has the form of a viscous damping coefficient. A model which incorporates its effect is shown below.



Each fluid grid point on the boundary is replaced by a pair of closely spaced grid points, between which a viscous damper, B_n, is connected for each harmonic order. One, but not both, of each pair of fluid grid points is identified as lying on the structural interface.

The above model neglects the velocity of the structure and also neglects the frictional force on the structure. These relatively small effects may be incorporated by means of additional direct matrix terms.

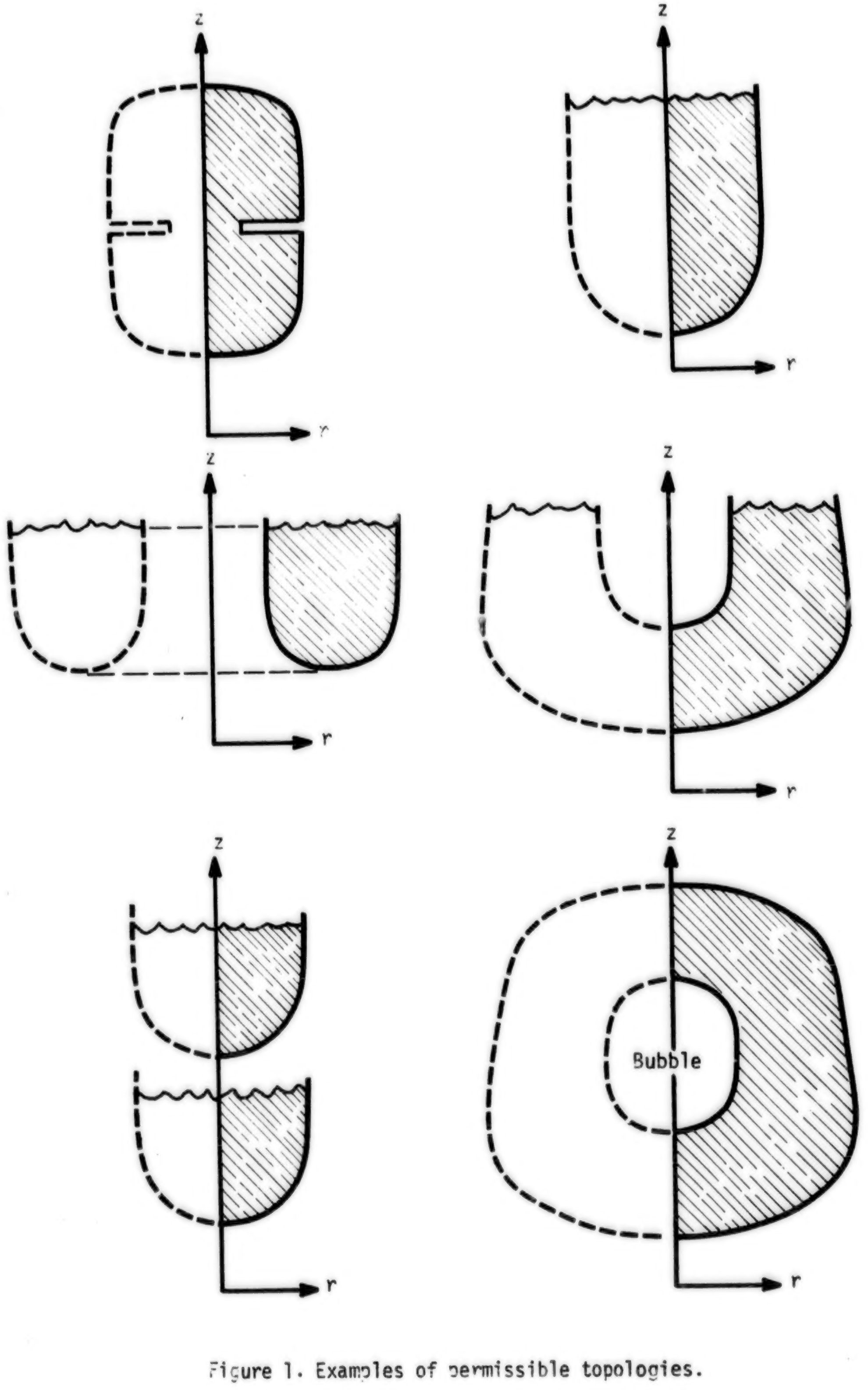

z
r
z
r
z
r
z
r
z
r
z
Bubble
r

Figure 1. Examples of permissible topologies.

16.2 COMPRESSIBLE FLUIDS IN ROTATIONALLY SYMMETRICAL CAVITIES WITH SLOTS

This section describes an application of NASTRAN to the acoustic analysis of cavities with the symmetry properties shown in Figure 1. The cavity is assumed to consist of an axisymmetric central region and two or more symmetrically arranged slotted extensions. The application does not consider interaction with the surrounding structure. Internal procedures and data cards have been designed to automate the computation of the vibration modes of the cavity. Additional information is included in the User's Manual and in Reference 1.

16.2.1 Formulation of the Problem

The general case of an unsymmetrical acoustic cavity may be solved with a finite element model representing the wave equation in three dimensions. In the case of symmetrical cavities, the problem may be reduced, depending on the actual symmetry and shape of the cavity. The various choices of simplification are:

1. If planes of symmetry exist in the cavity, a three dimension finite element model may be used to model a portion of the cavity and the acoustic resonance modes may be extracted in separate runs using different boundary conditions. Figure 2 shows an example of the use of symmetry to reduce the model to one-fourth size using two planes of symmetry.

2. If the shape of the cross section does not vary along the axis, the problem may be solved with a two dimensional model of the cross section. See Figure 3 for an illustration of this technique.

3. If the cavity is axisymmetric, i.e., if the outer radius is independent of the circumferential angle, the problem may be reduced to a two dimensional problem in r (the radius) and z (the axis). The motion is expressed as harmonics about the circumference and each harmonic is solved as a separate disjoint problem. This is the type of problem treated in Section 16.1.

PRECEDING PAGE BLANK NOT FILMED

PAGE 933 INTENTIONALLY BLANK

4. A special case exists for typical solid rocket motor cavities wherein the cross section contains a round central hole and symmetrically arranged radial slots as shown in Figure 1. If the slots are sufficiently narrow the gradient of pressure across them can be ignored; and if the slots are evenly spaced, the formulation of the problem in terms of harmonic degrees of freedom becomes relatively easy. The net result is that the vibrations within the slots can be modeled by harmonics. Thus both the slots and the central circular cavity reduce to two dimensional problems in harmonic analysis. This is the formulation that has been selected.

The theoretical treatment of the axisymmetric central region is identical to that described in Sections 16.1.2 and 16.1.4. The cross-section is divided into triangular and quadrilateral elements and special trapezoidal elements next to the axis. The slotted region is also divided into triangular and quadrilateral elements. The complete finite element model has the appearance shown in Figure 4. Note that axisymmetric tubular regions that have no direct connection to the central cavity may be attached to the slots.

The main subjects treated in following sections are the development of finite elements for the slot region, and the procedures used to interconnect the slots with the axisymmetric regions. The formulas required to recover velocity components for both types of regions are also described.

The slot elements can also be used by themselves to solve both static and dynamic two-dimensional potential problems including, in addition to acoustic problems, fluid flow, heat conduction, gravity waves in shallow water, electrical wave transmission, etc. The user can specify a different slot width at each gridpoint in the field.

16.2.2 Fluid Elements for the Slots

16.2.2.1 Energy Expressions

A slot region is defined as a series of narrow, evenly spaced cavities with their midplanes defined by planes passing through the axis of the fluid. If the pressure gradient across each slot is ignored, the pressure in the j^{th} slot can be described by a finite Fourier series:

$$p_j = p^o(r,z) + \sum_{n=1}^{N} p^n(r,z) \cos n\phi_j + \sum_{n=1}^{N^*} p^{n^*}(r,z) \sin n\phi_j \ . \qquad (1)$$

The slots are placed at angles

$$\phi_j = \frac{2\pi j}{M} \qquad j = 0,1,2,\ldots, M-1 \quad , \tag{2}$$

where M is the total number of slots. The total number of terms in the series, $1+N+N^*$, must equal M in order to provide a unique decomposition of an arbitrary pressure distribution into Fourier components. If M is an odd number

$$N = N^* = \frac{M-1}{2} \quad . \tag{3}$$

If M is an even number

$$\left. \begin{aligned} N &= \frac{M}{2} \quad , \\ N^* &= \frac{M}{2} - 1 \, . \end{aligned} \right\} \tag{4}$$

For this case $\sin\left(\frac{M}{2}\,\phi_j\right) = \sin(j\pi) = 0$, so that the sine coefficient of order $n^* = \frac{M}{2}$ provides no pressure and we may select the upper limit of N^* equal to $\frac{M}{2}$ if we wish.

The equation for generalized potential energy is the same as Equation 9 of Section 16.1.4, except that the integration over ϕ is carried out as follows:

$$\int_0^{2\pi} f(\phi) r \, d\phi = \sum_{j=0}^{M-1} w \, f(\phi_j) \quad , \tag{5}$$

where $f(\phi)$ is any function and w is the slot width assumed to be equal for all slots.

Substituting Equation 5 into Equation 9 of Section 16.1.4 results in the following expression for the potential energy:

$$U = \frac{1}{2} \int_A \left(\frac{w}{\rho} \sum_{j=0}^{M-1} \nabla p(\phi_j) \cdot \nabla p(\phi_j) \right) dA \quad . \tag{6}$$

where $dA = dr\,dz$.

Ignoring the gradient in the ϕ direction, the equation for the pressure gradient is:

$$\nabla p = \left[\frac{\partial p^o}{\partial r} + \sum_{n=1}^{K\le\frac{M}{2}}\left(\frac{\partial p^n}{\partial r}\cos n\phi + \frac{\partial p^{n*}}{\partial r}\sin n\phi\right)\right]\vec{e}_r$$

$$+ \left[\frac{\partial p^o}{\partial z} + \sum_{n=1}^{K\le\frac{M}{2}}\left(\frac{\partial p^n}{\partial z}\cos n\phi + \frac{\partial p^{n*}}{\partial z}\sin n\phi\right)\right]\vec{e}_z \quad , \qquad (7)$$

where K = M/2 for M even and K = (M-1)/2 for M odd.

Substituting Equation 7 into Equation 6 results in a rather involved expression. The cross product terms between different harmonics will, however, disappear, which may be shown as follows: Since $\phi = 0$ is a plane of symmetry:

$$\sum_{j=0}^{M-1}\cos\frac{2\pi mj}{M}\,\sin\frac{2\pi nj}{M} = 0 \qquad \text{all } m,n \quad . \qquad (8)$$

Define the coefficients:

$$R_{mn} = \sum_{j=0}^{M-1}\cos\frac{2\pi mj}{M}\,\cos\frac{2\pi nj}{M} = \frac{1}{2}\sum_{j=0}^{M-1}\left(\cos\frac{2\pi j}{M}(m-n) + \cos\frac{2\pi j}{M}(m+n)\right),$$

$$S_{mn} = \sum_{j=0}^{M-1}\sin\frac{2\pi mj}{M}\,\sin\frac{2\pi nj}{M} = \frac{1}{2}\sum_{j=0}^{M-1}\left(\cos\frac{2\pi j}{M}(m-n) - \cos\frac{2\pi j}{M}(m+n)\right). \quad (9)$$

The only conditions under which the sums of the cosine terms give a nonzero result occurs when $\frac{m+n}{N} = 0$ or 1, or when $\frac{n-m}{M} = 0$ or 1. The latter condition can occur only if n = m. Since $n \le M/2$ and $m \le M/2$ the former condition can occur only if m = n = 0 or if m = n = M/2. Thus all of the cross product terms $(m \ne n)$ disappear. In addition.

$$\left.\begin{aligned} R_{mm} &= S_{mm} = \frac{M}{2} \qquad \text{for } m \ne 0, \frac{M}{2} \quad , \\ R_{00} &= R_{\frac{M}{2},\frac{M}{2}} = M \quad , \\ S_{00} &= S_{\frac{M}{2},\frac{M}{2}} = 0 \quad . \end{aligned}\right\} \qquad (10)$$

Since the cross-product terms disappear, the potential energy may be written,

$$U = U^{o} + \sum_{n=1}^{K\le\frac{M}{2}} (U^{n} + U^{n*}) + U^{\frac{M}{2}} \quad , \tag{11}$$

where

$$\left.\begin{aligned}
U^{n} &= \frac{M}{4}\int_A \frac{w}{\rho}\,(\nabla_2 p^{n})\cdot(\nabla_2 p^{n})\;dA \qquad 0 < n < \frac{M}{2} \;, \\
U^{n*} &= \frac{M}{4}\int_A \frac{w}{\rho}\,(\nabla_2 p^{n*})\cdot(\nabla_2 p^{n*})\;dA \qquad 0 < n* < \frac{M}{2} \;, \\
U^{o} &= \frac{M}{2}\int_A \frac{w}{\rho}\,(\nabla_2 p^{o})\cdot(\nabla_2 p^{o})\;dA \quad , \\
U^{\frac{M}{2}} &= \frac{M}{2}\int_A \frac{w}{\rho}\,(\nabla_2 p^{M/2})\cdot(\nabla_2 p^{M/2})\;dA \quad ,
\end{aligned}\right\} \tag{12}$$

∇_2 is the two dimensional gradient operator

$$\nabla_2 = \vec{e}_r \frac{\partial}{\partial r} + \vec{e}_z \frac{\partial}{\partial z} \quad . \tag{13}$$

In an analysis of the vibration modes of the cavity, the results for the sine coefficients, p^{n*}, will be identical to the results for the cosine coefficients, and they are, therefore, of no further interest. Each distinct cosine coefficient, $0 \le n \le M/2$, produces a distinct set of vibration modes.

The kinetic energy may be analyzed in the same manner as the potential energy. From Equation 16 of Section 16.1.2

$$\left.\begin{aligned}
T^{n} &= \frac{M}{4}\int_A \frac{w}{B}\,(\dot{p}^{n})^2\;dA \qquad 0 < n < M/2 \;, \\
T^{o} &= \frac{M}{2}\int_A \frac{w}{B}\,(\dot{p}^{o})^2\;dA \quad , \\
T^{M/2} &= \frac{M}{2}\int_A \frac{w}{B}\,(\dot{p}^{M/2})^2\;dA \quad .
\end{aligned}\right\} \tag{14}$$

16.2.2.2 Triangular Slot Elements

The pressure distribution within each triangular slot element is assumed to be linear with respect to r and z and of the form:

$$p^n(r,z) = f_1(r,z)\, p_1^n + f_2(r,z)\, p_2^n + f_3(r,z)\, p_3^n \quad , \tag{15}$$

where p_j^n, j = 1, 2, 3, are the values of pressure at the corners.

Each term of the stiffness matrix is calculated as a second partial derivative of the energy,

$$K_{ij}^n = \frac{\partial^2 U^n}{\partial p_i^n\, \partial p_j^n} \quad . \tag{16}$$

Thus, from Equations 12, 15, and 16,

$$K_{ij}^n = \frac{M}{2} \int_A \frac{w}{\rho}\, \nabla_2 (f_i) \cdot \nabla_2 (f_j)\, dA \quad , \tag{17}$$

for $0 < n < \frac{M}{2}$. For n = 0, $\frac{M}{2}$, the stiffness matrix terms are twice as large. In an analogous manner the mass matrix terms for $0 < n < \frac{M}{2}$ are, using Equation 14,

$$M_{ij}^n = \frac{\partial^2 T}{\partial \dot{p}_i^n\, \partial \dot{p}_j^n} = \frac{M}{2} \int_A \frac{w}{B}\, f_i f_j\, dA \quad . \tag{18}$$

The functions f_i may be evaluated as follows. Observe that the expression

$$X(r,z) = \frac{\begin{vmatrix} 1 & r & z \\ 1 & r_2 & z_2 \\ 1 & r_3 & z_3 \end{vmatrix} X_1 + \begin{vmatrix} 1 & r_1 & z_1 \\ 1 & r & z \\ 1 & r_3 & z_3 \end{vmatrix} X_2 + \begin{vmatrix} 1 & r_1 & z_1 \\ 1 & r_2 & z_2 \\ 1 & r & z \end{vmatrix} X_3}{\begin{vmatrix} 1 & r_1 & z_1 \\ 1 & r_2 & z_2 \\ 1 & r_3 & z_3 \end{vmatrix}} \quad , \tag{19}$$

implies that X is a linear function with values X_1, X_2, and X_3 at positions (r_1, z_1), (r_2, z_2) and (r_3, z_3) respectively. f_1, f_2, and f_3 are thereby equal to the coefficients of X_1, X_2, and X_3

respectively in the equation above. The value of the denominator equals twice the cross sectional area.

The integrals given in Equations 17 and 18 are evaluated under the assumption that the compressibility, B, and the density, ρ, are constant over the element. The thickness, w, is assumed to vary linearly over the element. It may, therefore, be expressed by the equation:

$$w(r,z) = f_1(r,z)w_1 + f_2(r,z)w_2 + f_3(r,z)w_3 \quad . \tag{20}$$

The gradients are:

$$\nabla_2 f_i(r,z) = \frac{1}{2A}\left[(z_j - z_k)\,\vec{e}_r + (r_k - r_j)\,\vec{e}_z\right] \quad , \tag{21}$$

where j and k are the other two points in cyclic order and A is the area of the triangle.

Observing that the gradient of pressure is constant over the element, the stiffness matrix terms, obtained by inserting Equation 20 into Equation 17, are

$$K^n_{ij} = \frac{1}{2\rho}\,\nabla_2(f_i)\cdot\nabla_2(f_j)\int_A (f_1w_1 + f_2w_2 + f_3w_3)\,dA \quad . \tag{22}$$

Combining Equations 18 and 20 the mass matrix terms are:

$$M^n_{ij} = \frac{M}{2B}\int (w_1f_1 + w_2f_2 + w_3f_3)\,f_i\,f_j\,dA \quad . \tag{23}$$

The integral equations for stiffness and mass may be evaluated by the identity

$$\int_A f_i^{\alpha}\,f_j^{\beta}\,f_k^{\gamma}\,dA = \frac{\alpha!\;\beta!\;\gamma!}{(2+\alpha+\beta+\lambda)!}\,2A \quad , \tag{24}$$

where i, j and k are distinct and A is the area of the triangle.

Substituting Equation 24 repeatedly into Equations 22 and 23 results in stiffness and mass terms for the triangular slot element for [illegible] n < M/2 as follows:

$$\left.\begin{aligned} K^n_{ij} &= \frac{MA}{6\rho}(w_1 + w_2 + w_3)\left[\nabla_2(f_i)\cdot\nabla_2(f_j)\right] \;, \\ M^n_{ij} &= \frac{MA}{120B}(w_i + w_j + w_1 + w_2 + w_3)(1 + \delta_{ij}) \;, \end{aligned}\right\} \tag{25}$$

where:

$$\delta_{ij} = 1, \quad i = j \;,$$

$$\delta_{ij} = 0, \quad i \neq j \;,$$

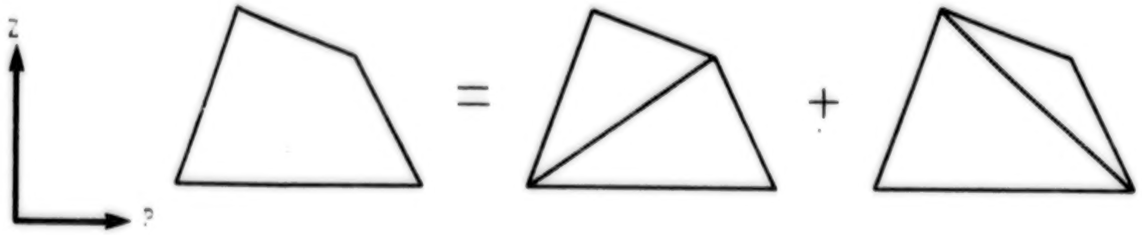
and

$$\nabla_2(f_i)\cdot\nabla_2(f_j) = \frac{1}{4A^2}\left[(z_j - z_k)(z_k - z_i) + (r_k - r_j)(r_i - r_k)\right] \;. \tag{26}$$

For n = 0, M/2, the values are twice as large.

16.2.2.3 Quadrilateral Slot Elements

The quadrilateral slot element consists of four triangular slot elements defined by the exterior edges and diagonals of the quadrilateral as shown below:



The points may be input in any order but each interior angle must be less than 180 degrees. The stiffness and mass matrices for the quadrilateral are taken as one-half the sums of stiffness and mass matrices for the individual triangles.

16.2.3 Corrections for the Effects of Finite Slot Width

16.2.3.1 Circumferential Variation of Pressure

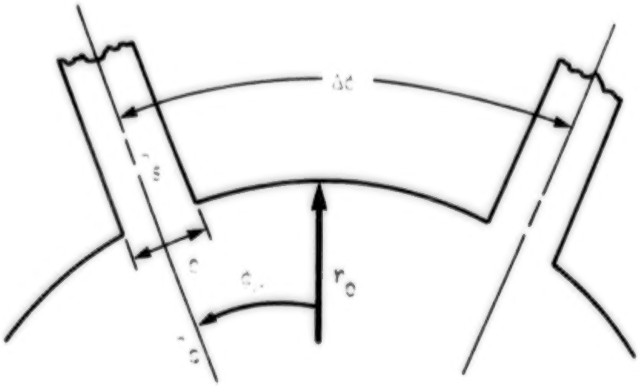
In the central cavity, it is assumed that the pressure

$$p_c = p_c^n(r,z)\cos n\phi \quad . \tag{1}$$

It is also assumed, in the slotted region, that, in the m^th slot,

$$p_{s,m} = p_s^n(r,z)\cos n\phi_m \quad . \tag{2}$$

where ϕ_m is the azimuth angle at the midplane of the m^th slot, see sketch below. A question then arises as to how the pressure at the mouth of the slot is related to the pressure in the cylindrical region.



We might, for example, assume that the pressure in the mouth of the m^th slot, $p_{s,m}$, is equal to the pressure in the cylindrical region evaluated at $\phi = \phi_m$, the midpoint of the slot. A better assumption, particularly for wide slots, is that $p_{s,m}$ is equal to the average pressure in the cylindrical region, averaged over the slot width. Thus, using Equation 1, the pressure in the width of the m^th slot is

$$p_{s,m} = p_c^n(r_o,z)\,\frac{1}{w_o}\int_{\phi_m-\frac{w}{2r_o}}^{\phi_m+\frac{w}{2r_o}} \cos n\phi\cdot r_o\, d\,\phi$$

$$= p_c^n(r_o,z)\cos n\phi_m \cdot \frac{\sin\frac{nw_o}{2r_o}}{\frac{nw_o}{2r_o}}$$

$$= p_{c,m}\cdot\frac{\sin\frac{nw_o}{2r_o}}{\frac{nw_o}{2r_o}} \quad . \tag{3}$$

Note that the ratio $p_{s,m}/p_{c,m}$ is independent of m so that, upon substituting Equations 1 and 2 into Equation 3,

$$\frac{p_s^n(r_o,z)}{p_c^n(r_o,z)} = \frac{\sin\frac{nw_o}{2r_o}}{\frac{nw_o}{2r_o}} \quad . \tag{4}$$

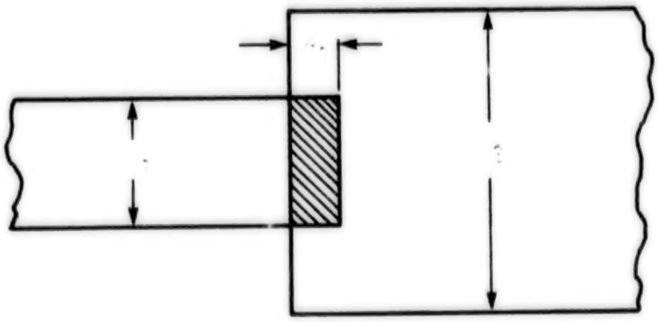
The factor $\frac{nw_o}{2r_o}$ achieves a maximum value of $\pi/2$ for $n = M/2$ (the highest mode index) and $w_o = 2\pi r_o/M$, i.e., if the slots occupy the entire circumference. The following table shows how the ratio of p_s^n to p_c^n depends on $nw_o/2r_o$

$\frac{nw_o}{2r_o}$	0	.2	.4	.6	.8	1.0	$\pi/2$
$\frac{p_s^n}{p_c^n}$	1.0	.995	.975	.942	.898	.840	.636

Note that the effect becomes significant for $nw_o/2r_o > 0.5$, i.e., when each slot covers greater than a twelfth of a wave length.

16.2.3.2 Effect of Sharp Corners

It may be shown that the effect of an abrupt change in the diameter of a pipe is to add a coupling impedance that is equivalent to a short additional length of pipe (Reference 2). The problem here is a little different in that the slot is long (in the z direction) compared to its width. Smythe (Reference 3) has solved the analogous problem for the added resistance of a conducting strip whose width suddenly changes. His solution may easily be expressed in terms of the added effective length, ℓ_e, of the strip whose dimensions are shown below.



The impedance of the added length is

$$K_e = \frac{w_1 t}{\rho \ell_e} \quad , \tag{5}$$

where ρ is the density of the fluid and t is the transverse dimension of the slot in a direction normal to the plane of the figure. Smythe's expression for ℓ_e is

$$\ell_e = \frac{h}{\pi}\left[\frac{(h+K)^2}{hK}\log_e(K+h) - \frac{(K-h)^2}{hK}\log_e(K-h) - 2\log_e 4hK\right] \quad , \tag{6}$$

where $h = w_1/2$ and $K = w_2/2$. Manipulation of Equation 6 leads to

$$\ell_e = \frac{w_1}{2\pi}\left[(\beta + \frac{1}{\beta})\log_e\frac{\beta+1}{\beta-1} + 2\log_e\frac{(\beta+1)(\beta-1)}{4\beta}\right] \quad , \tag{7}$$

where $\beta = w_2/w_1$.

Equation 7 is plotted in Figure 5. The bracketed expression has a logarithmic singularity as $\beta \to \infty$, but it is seen that $\ell_e/\sqrt{w_1 w_2}$ remains finite (in fact approaches zero) as $\beta \to \infty$. By way of comparison, the effective length of a round pipe of diameter w_1 approaches $4w_1/3\pi = .42w_1$ as $\beta \to \infty$.

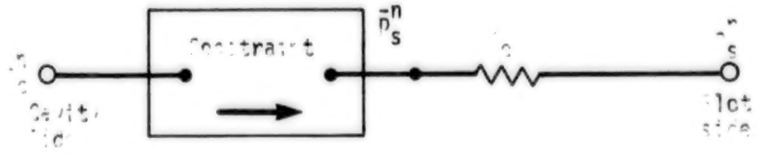
In our application

$$\left.\begin{aligned} w_1 &= \text{width of slot mouth, } w_o \quad , \\ w_2 &= 2\pi r_o/M \quad , \\ K_e &= \frac{Mw_o t}{2\rho\ell_e} \quad , \qquad 0 < n < \frac{M}{2} \quad . \end{aligned}\right\} \tag{8}$$

For $n = 0, \frac{M}{2}$, K_e is twice as large; t is the length of slot in the (r,z) plane associated with the grid point.

16.2.3.3 Combination of the Two Effects

The effects described in the two preceding subsections may be combined to define a stiffness matrix that couples p_s^n and p_c^n. The two effects may be represented schematically as follows



The equation of constraint is

$$\bar{p}_s^n = p_c^n \frac{\sin \frac{nw_o}{2r_o}}{\frac{nw_o}{2r_o}} \quad . \tag{9}$$

The scalar spring is given by Equation 8. The equation of constraint and the scalar spring produce a "stiffness" matrix as follows

$$\begin{Bmatrix} F_s \\ \hline F_c \end{Bmatrix} = \left[\begin{array}{c|c} K_e & -K_e\alpha \\ \hline -K_e\alpha & K_e\alpha^2 \end{array}\right] \begin{Bmatrix} p_s^n \\ \hline p_c^n \end{Bmatrix} , \tag{10}$$

where

$$\alpha = \frac{\sin \dfrac{nw_o}{2r_o}}{\dfrac{nw_o}{2r_o}} \tag{11}$$

In the finite element model, Figure 4, the grid points on the slot boundary each have two degrees of freedom, p_s^n and p_c^n, which are coupled by the stiffness matrix in Equation 10. All other grid points have only one degree of freedom. In the limiting case of sharp corners between slots, $w_2/w_1 = 1.0$, so that the effective length, ℓ_e, is zero and the stiffness K_e is infinite. In order to avoid recoding for this special case, the effective length is chosen to be not less than one-hundredth of the width of the slot.

16.2.4 Recovery of Velocity Components

The velocity components within the fluid are calculated in the stress recovery phase of NASTRAN. Using the momentum relation, Equation 1a of Section 16.1.2,

$$\vec{\ddot{u}} = -\frac{1}{\rho}\nabla p \quad . \tag{1}$$

Since the frequency is known in the cases of interest, the velocity is:

$$\vec{\dot{u}} = -\frac{1}{i\omega\rho}\nabla p \quad , \tag{2}$$

where ω is the radian frequency and i is the imaginary number indicating a phase shift of 90 degrees. In normal modes analysis, the velocity is considered to be composed of real numbers and the phase shift is ignored. In frequency response analysis the velocity will be given as a complex number. In complex eigenvalue analysis, the complex eigenvalue $p = i\omega + \sigma$ is used instead of $i\omega$.

The velocities in a triangular fluid element at any point (r,z) in the axisymmetric region are:

$$V_r = -\frac{1}{i\omega\rho}\frac{\partial p}{\partial r} = -\frac{1}{i\omega\rho} q_1^n \quad , \tag{3}$$

$$V_\phi = -\frac{n}{i\omega\rho r_0}\frac{\partial p}{\partial \phi} = -\frac{n}{i\omega\rho r}\left(q_0 + q_1^n r + q_2^n z\right) , \tag{4}$$

$$V_z = -\frac{1}{i\omega\rho}\frac{\partial p}{\partial z} = -\frac{1}{i\omega\rho} q_2^n \quad , \tag{5}$$

where q_i^n are the generalized coordinates described in Section 16.1.4.2. In the elements for the slotted region, V_r and V_z have the same formulas, but V_ϕ is zero. The velocities are evaluated at four points, the centroid and the midpoints of the outer edges. The velocity in the circumferential direction, V_ϕ, is the value that occurs at $\phi = -\frac{\pi}{2n}$, $n \neq 0$. The other values occur at $\phi = 0$.

At the centroid:

$$\begin{Bmatrix} V_r \\ V_\phi \\ V_z \end{Bmatrix} = -\frac{1}{i\omega\rho} \begin{bmatrix} 0 & 1 & 0 \\ \frac{n}{r_c} & n & \frac{nz_c}{r_c} \\ 0 & 0 & 1 \end{bmatrix} [H_{qp}^n] \begin{Bmatrix} p_1^n \\ p_2^n \\ p_3^n \end{Bmatrix} \quad , \tag{6}$$

where

$$\begin{aligned} r_c &= \frac{1}{3}\left(r_1 + r_2 + r_3\right) \quad , \\ z_c &= \frac{1}{3}\left(z_1 + z_2 + z_3\right) \quad , \end{aligned} \tag{7}$$

and $[H_{qp}^n]$ is the matrix which transforms pressures, p_i^n, to generalized coordinates, q_i^n, see Equation 19 of Section 16.1.4.2.

At the midpoint of each edge:

$$\left.\begin{aligned} V^s_{ij} &= -\frac{1}{i\omega\rho}\,\frac{p_j - p_i}{\ell_{ij}} \,, \\ V^\phi_{ij} &= -\frac{n}{i\omega\rho}\,\frac{p_i + p_j}{r_i + r_j} \,, \\ \text{where} \quad \ell_{ij} &= \sqrt{(r_i - r_j)^2 + (z_i - z_j)^2} \,, \end{aligned}\right\} \qquad (8)$$

These formulas are derived using the assumption that the pressure is a linear function of position. V^s_{ij} is the component of velocity directed along the edge of the element in the r,z plane (at $\phi = 0$). V^ϕ_{ij} is the velocity in the circumferential direction (at $\phi = -\pi/2n$).

The velocities of the quadrilateral fluid element are calculated at the intersection of the diagonals by averaging the velocities of each of the four subtriangles. The velocities along each edge are calculated using the same equations as with the triangular fluid element.

The velocities in the trapezoidal fluid elements next to the axis of symmetry are calculated at the center (r = 0) and along the outer edge. The velocity at the center has nonzero values only for n = 0 and n = 1. At the center:

$$\left.\begin{aligned} V_r &= -\frac{1}{i\omega\rho}\,\frac{(p_1 + p_2)}{(r_1 + r_2)} && n = 1 \,, \\ V_r &= 0 && n \neq 1 \,, \\ V_z &= -\frac{1}{i\omega\rho}\,\frac{(p_2 - p_1)}{(z_2 - z_1)} && n = 0 \,, \\ V_z &= 0 && n \neq 0 \,. \end{aligned}\right\} \qquad (9)$$

At the outer edge:

$$\left.\begin{aligned} V_s &= \frac{-1}{i\omega\rho}\left[\frac{\partial p^n}{\partial r}\,\frac{(r_2 - r_1)}{\ell} + \frac{\partial p^n}{\partial z}\,\frac{(z_2 - z_1)}{\ell}\right] \quad , \\ V_\phi &= \frac{-n}{i\omega\rho r}\,p^n \quad , \\ \text{where} \qquad \ell &= \sqrt{(r_2 - r_1)^2 + (z_2 - z_1)^2} \quad . \end{aligned}\right\} \qquad (10)$$

Since the velocities are calculated at the midpoint of the outer edge, the terms are evaluated at:

$$\left.\begin{aligned} \bar{r} &= \frac{r_1 + r_2}{2} \quad , \\ \bar{z} &= \frac{z_1 + z_2}{2} \quad . \end{aligned}\right\} \qquad (11)$$

The derivatives of the pressure at the midpoint of the outer edge are:

$$\left.\begin{aligned} \frac{\partial p}{\partial r} &= \frac{n\bar{r}^n}{\bar{r}}\,(q_1^n + q_2^n z) \quad , \\ \frac{\partial p}{\partial z} &= \bar{r}^n\,q_2^n \quad , \end{aligned}\right\} \qquad (12)$$

where the generalized coordinates at points (1) and (2) are related to the pressure coefficients by,

$$\begin{Bmatrix} q_1^n \\ q_2^n \end{Bmatrix} = [H_{qp}^n] \begin{Bmatrix} p_1^n \\ p_2^n \end{Bmatrix} \quad . \qquad (13)$$

See Equation 45 of Section 16.1.4.4 for evaluation of $[H_{qp}^n]$.

Evaluating Equation 11 through 13 results in the velocity vector along the outer edge:

$$\begin{Bmatrix} V_s \\ \hdashline V_\phi \end{Bmatrix} = \frac{-\bar{r}^{\,n-1}}{i\omega\rho}\left[\begin{array}{c|c} \dfrac{n(r_2-r_1)}{\ell} & \dfrac{n\bar{z}(r_2-r_1) + \bar{r}(z_2-z_1)}{\ell} \\ \hline n & n\bar{z} \end{array}\right] [H_{qp}^n] \begin{Bmatrix} p_1^n \\ \hdashline p_2^n \end{Bmatrix} \quad . \qquad (14)$$

REFERENCES

1. Herting, D. N., Joseph, J. A., Kuusinen, L. R., and MacNeal, R. H., "Acoustic Analysis of Solid Rocket Motor Cavities by a Finite Element Method," Air Force Rocket Propulsion Laboratory, Report No. AFRPL TR 71-96, 1971.

2. Rschevkin, S. N., The Theory of Sound, The MacMillan Co., New York, 1963, p. 218.

3. Smythe, W. R., Static and Dynamic Electricity, McGraw-Hill Book Co., Inc., New York, 1939, pp. 230-233.

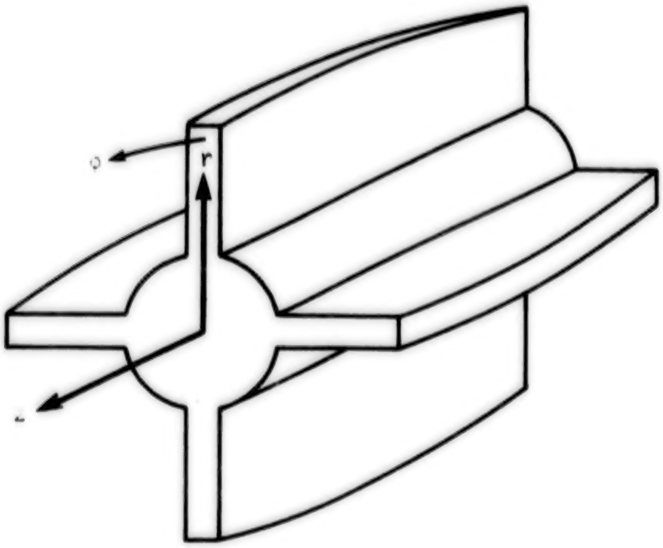
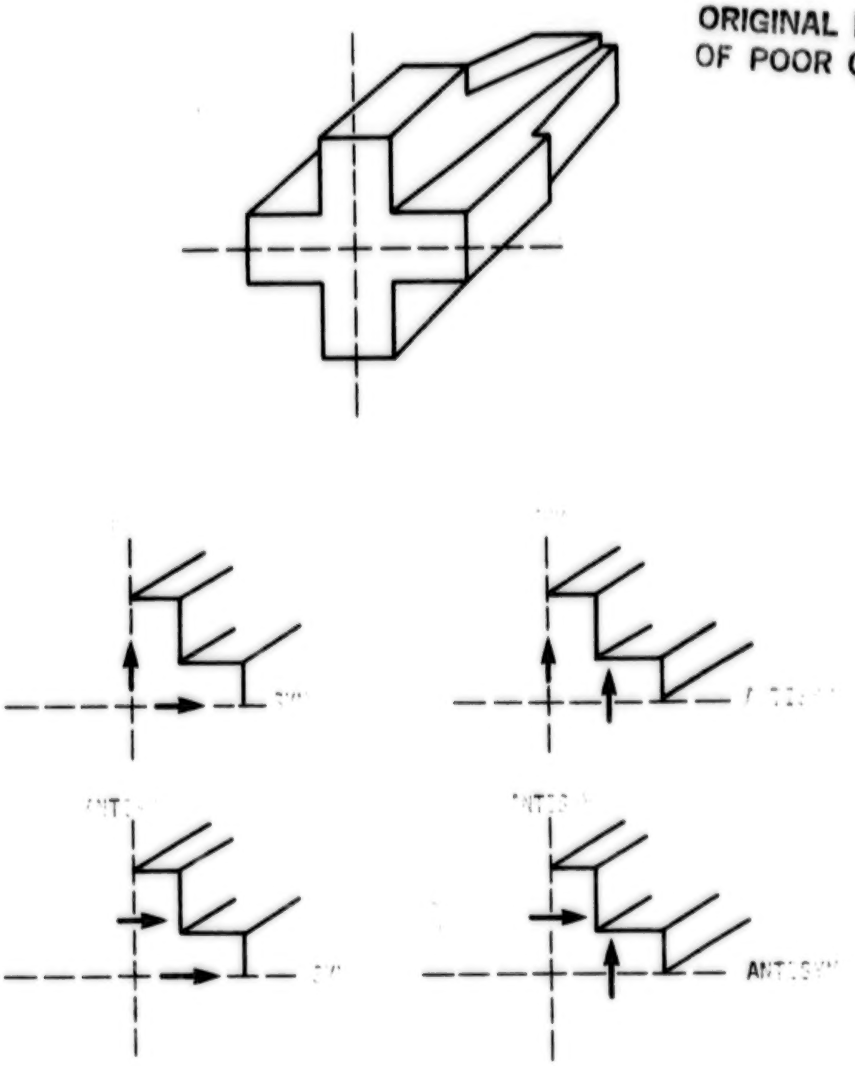


Figure 1. Symmetrical acoustic cavity with slots.


SYM
ANTISYM
ANTISYM
ANTISYM
SYM
ANTISYM

Arrows (↑) denote flow direction at boundary

SYM = Symmetric Boundary

ANTISYM = Antisymmetric Boundary

Figure 2. Boundary conditions for one-quarter symmetry.

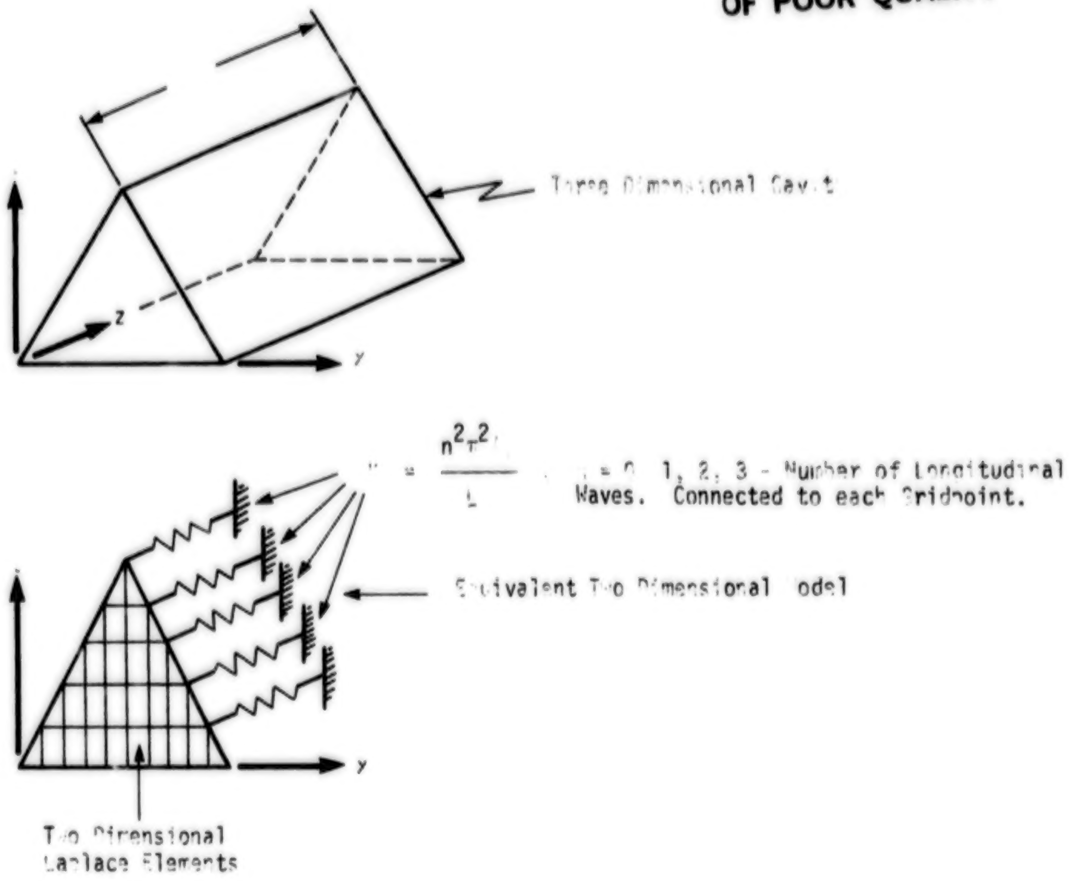

Three Dimensional Cavity
z
y
$= \frac{n^2\pi^2}{L}$
n = 0, 1, 2, 3 - Number of Longitudinal Waves. Connected to each Gridpoint.
Equivalent Two Dimensional Model
y
Two Dimensional Laplace Elements

Figure 3. Two dimensional representation of a three dimensional problem.

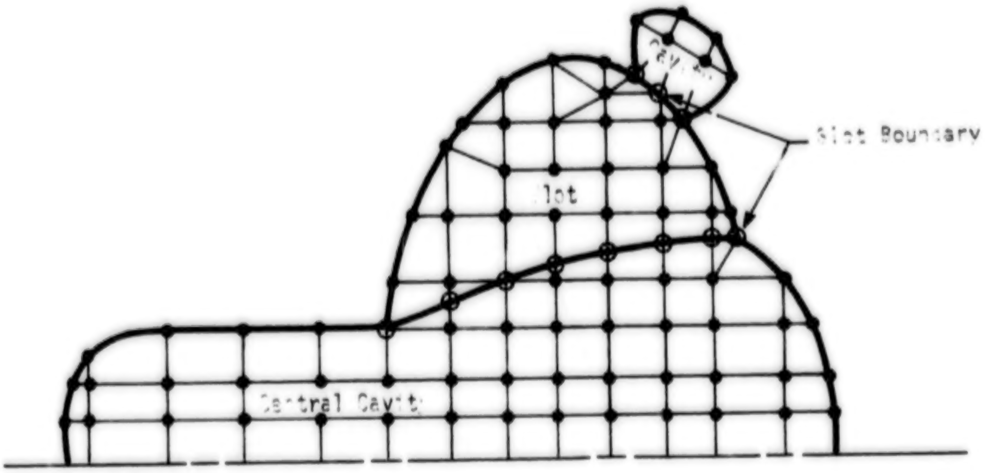

Cavity
Slot Boundary
Slot
Central Cavity

Figure 4. Finite element model of acoustic cavity.

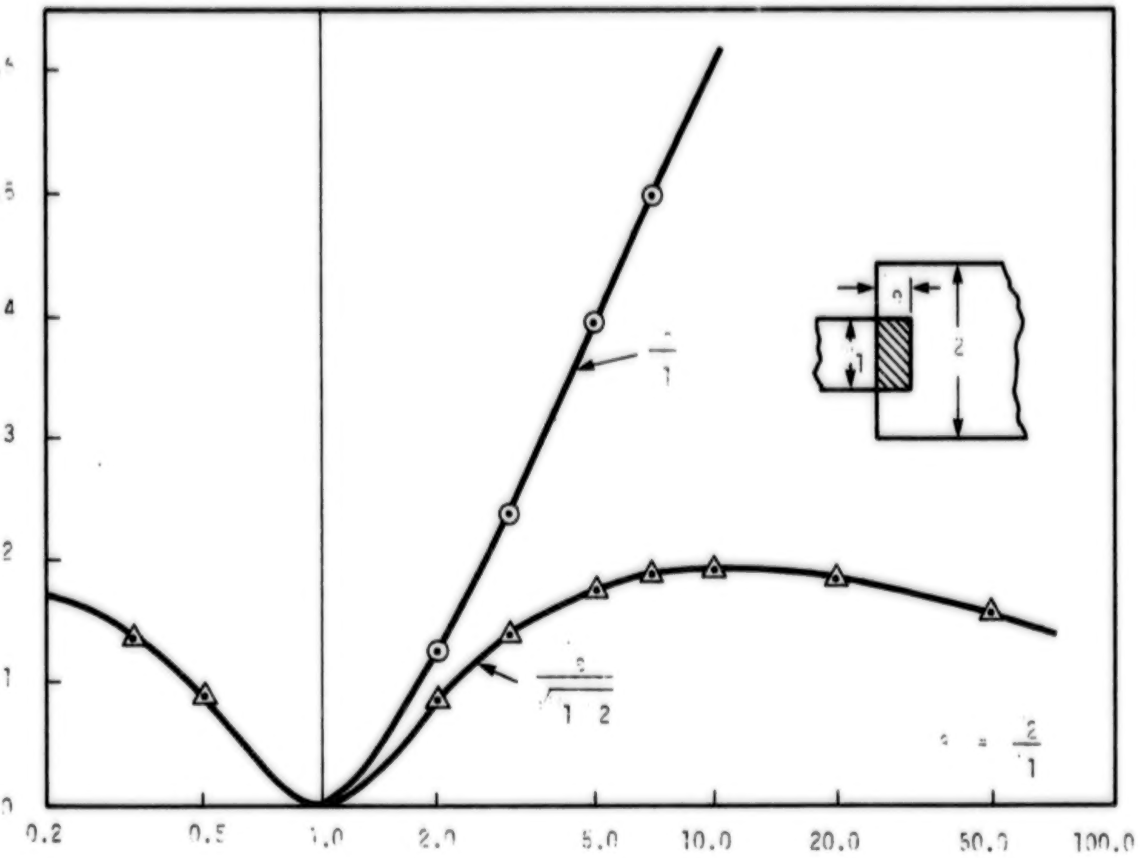


Figure 5. Effective length of an abrupt change in the width of a strip.

17. AEROELASTIC ANALYSIS

17.1 INTRODUCTION

Aeroelastic analysis brings a whole new range of capability to NASTRAN, which can be used with the structural analysis. Modules have been introduced to:

(a) Generate aerodynamic grid points,

(b) Provide connection (interpolation) between the structure and aerodynamics,

(c) Compute aerodynamic matrices, and

(d) Solve the equations for flutter or response.

Existing modules have been modified to provide, for example, a new method of eigenvalue extraction, the ability to plot aerodynamic elements and to plot the curves produced. Two rigid formats are available: one is for modal flutter analysis (by three methods, namely, the K-method, the KE-method and the PK-method) and the other is for modal transient and frequency response analyses of aeroelastic models.

Aerodynamic analysis, like structural analysis, is based upon a finite element approach. The finite elements are strips or boxes for which there are aerodynamic forces. There are two major points to be considered. The aerodynamic elements, even for rather complex vehicles, tend to be in regular arrays. Thus, while NASTRAN has the provision to generate arrays of structural elements, it is also desirable to generate arrays of aerodynamic elements. In particular, the aerodynamic elements for lattice methods are arrays of trapezoidal boxes whose sides are parallel to the airflow. These should be described simply by defining properties of the array (panel). The grid points defining the structure usually will not coincide with the grid points defining the aerodynamic elements. Provision has been made to generate equations of constraint between the two sets of grid points. The geometry interpolation is a key feature, since it allows the choice of structural and aerodynamic elements to be based upon structural and aerodynamic considerations separately.

Aerodynamic forces are generated via the flow surrounding the structure. The theory produces a matrix defining the forces upon the structure in terms of the deflections of the structure. State-of-the-art methods which involve interactions between aerodynamic elements are available only for sinusoidal motion. Phase lags occur between the motions and the forces, thus the matrices are complex. Furthermore, these complex matrices depend upon parameters of the flow, namely reduced frequency (ratio of frequency to velocity) and Mach number (ratio of velocity to speed of sound). Such a matrix, if computed by an interaction theory such as the doublet lattice method, will be expensive to produce. The most effective method to evaluate the matrix for a large number of parameter values is to compute the matrix for a few selected ones, and then interpolate to others.

This parametric interpolation is an automatic feature of the solution modules for modal aeroelastic analyses.

17.2 DATA INPUT AND GENERATION

Aerodynamic elements are strips, boxes or bodies which represent the structure for computation of aerodynamic forces. These elements, like the structural elements, are defined by their geometry, and their motions are defined by degrees of freedom at associated aerodynamic grid points. Often requirements of the aerodynamic theory will dictate the geometry of the boxes. For example, the doublet lattice theory requires trapezoidal boxes with their parallel edges along the streamlines. Also, if one box lies in the wake of another, their edges must be on the same two streamlines. Aerodynamic elements and grid points will be generated to reduce the labor of the user (many less data cards will be required) and to help insure that theoretical requirements are met.

Aerodynamic calculations are made in a cartesian aerodynamic coordinate system. By the usual convention the flow is in the positive x direction. The basic coordinate system was not chosen, since it would place an undesirable restriction upon the description of the structural model. Any NASTRAN cartesian system may be specified, and flow will be defined in the direction of its positive x axis. All aerodynamic calculations are made initially in the aerodynamic coordinate system. All element and aerodynamic grid point data, computed initially in the basic coordinate system, will be converted to the aerodynamic coordinate system. The global (displacement) coordinate system of the aerodynamic grid points will have its T1 direction in the flow direction. T3 is normal to the element for boxes, and parallel to the aerodynamic Z direction in the case of bodies. Coordinate system data are generated for the aerodynamic grid points.

The grid points are physically located at the centers of the boxes and body elements. Permanent constraints are generated for the unused degrees of freedom. A second set of grid points, used only for undeformed plotting, is located at the element corners. All six degrees of freedom associated with each grid point in this second set are permanently constrained. Grid point numbers are generated based upon the element identification number. For any panel, the external grid point numbers for the boxes start with the panel identification number and increase consecutively.

Aerodynamic degrees of freedom, along with the extra points, are added after the structural matrices and modes have been determined. This introduces the following displacement sets:

u_k	Aerodynamic box and body degrees of freedom
u_{sA}	Permanently constrained degrees of freedom associated with aerodynamic grid points
u_{ps}	Union of u_p (physical) and u_{sA}
u_{pA}	Union of u_k and u_{ps} (physical and aerodynamic).

The set u_{pA} replaces u_p as the set available for output at grid, scalar and extra points.

17.3 INTERCONNECTION OF STRUCTURE WITH AERODYNAMICS

Structural and aerodynamic degrees of freedom will be constrained by interpolation. This feature allows the selection of grid points and elements for the structure and for the aerodynamics to each be chosen in a method which is well suited to the particular problem. The structural model for a wing may involve a one-, two- or three-dimensional array of grid points. The aerodynamic theory may involve a strip theory, or a lifting surface theory. A general interpolation method is available which can interconnect the various combinations. Any aerodynamic panel or body can be subdivided into subzones for interpolation, using a separate function for each.

The interpolation method will be "splining." The theory involves the mathematical analysis of beams and plates (see Figure 1). Linear splines are a generalization of the simple beam, which allows torsional as well as bending degrees of freedom. Surface splines are solutions for infinite uniform plates. Several splines, including combinations of the two types, can be used in one model. For example, a model may use one linear spline for the horizontal tail, and three surface splines for the wing (inboard section, outboard section, and aileron). This local splining allows discontinuous slopes (for wing-aileron hinge), separate functions (for wing and tail), and smaller zones. Smaller zones will reduce the computing effort (cost).

The structural degrees of freedom have been chosen as the independent degrees of freedom; the aerodynamic degrees of freedom are dependent. A matrix is derived which relates the dependent degrees of freedom to the independent ones. The structural degrees of freedom may include any grid components.

17.3.1 Theory for Surface Splines

A surface spline is a mathematical tool used to find a function $w(x,y)$ for all points (x,y) when w is known for a discrete set of points, $w_i = w(x_i,y_i)$. A linear spline is a "beam" function which passes through the known points. The natural extension to two dimensions is to introduce an infinite plate, and solve for its deflection, given its deflection at a discrete set of points. This surface spline is a smooth continuous function which will become nearly linear in x and y at large distances from the points (x_i,y_i). Furthermore, the problem can be solved in closed form involving nothing more difficult than to evaluate some logarithm functions.

The deflection of the plate will be synthesized as the response due to a set of point loads on the infinite plate. The response due to a single load is called a fundamental solution. The fundamental solutions have polar symmetry. If the load is taken at $x_i = y_i = 0$, and polar coordinates are used ($x = r \cos \theta$, $y = r \sin \theta$), the governing differential equation is

$$D \nabla^4 W = D \frac{1}{r} \frac{d}{dr} \left\{ r \frac{d}{dr} \left[\frac{1}{r} \frac{d}{dr} \quad r \frac{dw}{dr} \right] \right\} = q \quad . \tag{1}$$

The load q vanishes except near r = 0. A solution to the general spline problem, formed by superimposing solutions of Equation 1 is given by

$$w(x,y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{N} K_i(x,y) P_i \quad , \tag{2}$$

where $K_i(x,y) = (1/16\pi D) r_i^2 \ln r_i^2$, $\qquad r_i = (x-x_i)^2 + (y-y_i)^2 \quad ,$

and $\qquad P_i$ = concentrated load at (x_i, y_i).

The N+3 unknowns (a_0, a_1, a_2, P_i, i=1,N) are determined from the N+3 equations

$$\Sigma P_i = \Sigma x_i P_i = \Sigma y_i P_i = 0, \text{ and}$$

$$w_j = a_0 + a_1 x_j + a_2 y_j + \sum_{i=1}^{N} K_{ij} P_i \quad (j=1,N) \quad , \tag{3}$$

where $K_{ij} = K_i(x_j, y_j)$.

Note that $K_{ij} = K_{ji}$, and $K_{ij} = 0$ when i = j. The details of the derivation are given in Reference 1.

These equations can be summarized in matrix form:

$$w(x,y) = [1, x, y \mid K_1(x,y), K_2(x,y), \ldots, K_N(x,y)] \left\{ \begin{array}{c} a_0 \\ a_1 \\ a_2 \\ \hline P_1 \\ P_2 \\ \vdots \\ P_N \end{array} \right\} \quad , \tag{4}$$

where $K_i(X,Y)$ is defined below Equation 2.

The vector of a's and p's is found by solving

$$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ \hline w_1 \\ w_2 \\ \vdots \\ w_N \end{Bmatrix} = \left[\begin{array}{ccc|cccc} 0 & 0 & 0 & 1 & . & . & 1 \\ 0 & 0 & 0 & x_1 & . & . & x_N \\ 0 & 0 & 0 & y_1 & . & . & y_N \\ \hline 1 & x_1 & y_1 & 0 & . & . & K_{1N} \\ 1 & x_2 & y_2 & . & . & . & K_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & K_{N1} & . & . & 0 \end{array}\right] \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ P_1 \\ P_2 \\ \vdots \\ P_N \end{Bmatrix} , \tag{5}$$

where K_{ij} is defined below Equation 3.

The interpolation to any point in the plane (x,y) is then achieved by evaluating $w(x,y)$ from Equation 2. Slopes can be found by analytic differentiation of Equation 2.

17.3.2 Theory of Linear Splines

Linear splines are easily solved by the three-moment method, which is excellent for simple linear splines. Unfortunately, the method does not work as well for splines with torsion, rigid arms and attachment springs. The derivations sketched below are based upon an analogy with the surface spline derivation.

a. Linear splines

Equation:

$$EI \frac{d^4w}{dx^4} = q - \frac{dM}{dx} , \tag{6}$$

where q = applied load and M = applied moment.

A symmetric fundamental solution for $x \neq 0$ is used for loads $q = P\delta(x)$, and an antisymmetric fundamental solution is used for moments. The solution for the general case is found by superimposing the fundamental solutions,

$$w(x) = a_0 + a_1 x + \sum_{i=1}^{N} \left(- \frac{M_i (x-x_i)|x-x_i|}{4EI} + \frac{P_i |x-x_i|^3}{12EI} \right) , \tag{7}$$

$$\theta(x) = \frac{dw}{dx} = a_1 + \sum_{i=1}^{N} \left(- \frac{M_i |x-x_i|}{2EI} + \frac{P(x-x_i)|x-x_i|}{4EI} \right) . \tag{8}$$

These are written in matrix notation as

$$\begin{Bmatrix} w(x) \\ \hline \theta(x) \end{Bmatrix} = \begin{bmatrix} 1 & x & \frac{|x-x_1|^3}{12EI} & \cdots & -\frac{(x-x_1)|x-x_1|}{4EI} & \cdots \\ 0 & 1 & \frac{(x-x_1)|x-x_1|}{4EI} & \cdots & -\frac{|x-x_1|}{2EI} & \cdots \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ P_1 \\ \vdots \\ P_N \\ M_1 \\ \vdots \\ M_N \end{Bmatrix} . \quad (9)$$

The unknowns a, P and M are found from

$$\begin{Bmatrix} 0 \\ 0 \\ \hline W_1 \\ \vdots \\ W_N \\ \theta_1 \\ \vdots \\ \theta_N \end{Bmatrix} = \left[\begin{array}{c|c|c} 0 & R_1^T & R_2^T \\ \hline R_1 & A_{11} & A_{21}^T \\ \hline R_2 & A_{21} & A_{22} \end{array}\right] \begin{Bmatrix} a_0 \\ a_1 \\ P_1 \\ \vdots \\ P_N \\ M_1 \\ \vdots \\ M_N \end{Bmatrix} , \quad (10)$$

where it has been assumed $x_1 < x_2 \ldots < x_N$, and

$$R_1^T = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_N \end{bmatrix}$$

$$R_2^T = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 1 \end{bmatrix}$$

$$A_{11} = \begin{bmatrix} 0 & \frac{(x_2-x_1)^3}{12EI} & \cdots & \frac{(x_N-x_1)^3}{12EI} \\ \frac{(x_2-x_1)^3}{12EI} & 0 & \cdots & \frac{(x_N-x_2)^3}{12EI} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{(x_N-x_1)^3}{12EI} & \frac{(x_N-x_2)^3}{12EI} & \cdots & 0 \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} 0 & -\frac{(x_2-x_1)^2}{4EI} & \cdots & -\frac{(x_N-x_1)^2}{4EI} \\ \frac{(x_2-x_1)^2}{4EI} & 0 & \cdots & -\frac{(x_N-x_2)^2}{4EI} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{(x_N-x_1)^2}{4EI} & \frac{(x_N-x_2)^2}{4EI} & \cdots & 0 \end{bmatrix}$$

$$A_{22} = \begin{bmatrix} 0 & -\frac{(x_2-x_1)}{2EI} & \cdots & -\frac{(x_N-x_1)}{2EI} \\ -\frac{(x_2-x_1)}{2EI} & 0 & \cdots & -\frac{(x_N-x_2)}{2EI} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{(x_N-x_1)}{2EI} & -\frac{(x_N-x_2)}{2EI} & \cdots & 0 \end{bmatrix}$$

b. Torsion Bars

Equation:

$$GJ\,\frac{d^2\theta}{dx^2} = -T \quad . \tag{11}$$

The solution is

$$\theta(x) = \left\lfloor 1 \;\middle|\; 0,\ -\frac{|x_2-x_1|}{2GJ},\ \ldots,\ -\frac{|x_N-x_1|}{2GJ} \right\rfloor \begin{Bmatrix} a_0 \\ \hline T_1 \\ \vdots \\ T_N \end{Bmatrix} \quad , \tag{12}$$

where the unknowns come from

$$\begin{Bmatrix} 0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_N \end{Bmatrix} = \left[\begin{array}{c|cccc} 0 & 1 & 1 & \cdots & 1 \\ \hline 1 & 0 & -\frac{|x_2-x_1|}{2GJ} & \cdots & -\frac{|x_N-x_1|}{2GJ} \\ 1 & -\frac{|x_2-x_1|}{2GJ} & 0 & \cdots & -\frac{|x_N-x_2|}{2GJ} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & -\frac{|x_N-x_1|}{2GJ} & -\frac{|x_N-x_2|}{2GJ} & \cdots & 0 \end{array} \right] \begin{Bmatrix} a_0 \\ \hline T_1 \\ T_2 \\ \vdots \\ T_N \end{Bmatrix} \quad . \tag{13}$$

17.3.3 Attachment of Splines with Elastic Springs

The change in the formulas for splines to accommodate the springs is very easy. A derivation, valid for the several types of splines, is as follows: The spline deflection is given by Equations 4, 9 or 12 and can be written

$$u_k(r) = [R(r)]\{a\} + [A_j(r)]\{P\} \quad , \tag{14}$$

where u_k is the deflection of the spline and the r may be a one- or two-dimensional argument. Thus, including the equilibrium equations 5, 10 or 13

$$0 = [R_i]^T\{P\} \quad , \tag{15}$$

and

$$\{u_k\} = [R_i]\{a\} + [A_{ij}]\{P\} \quad . \tag{16}$$

The structural deflection, u_g, will differ from the spline deflection by the deformation of the spring, resulting in forces

$$\{P\} = [K_s]\{u_g - u_k\} \quad , \qquad (17)$$

where the matrix, K_s, has the spring constant, K, along the diagonal. These are nonzero (if K were = 0, then there would be no attachment and that grid point would be discarded) and thus the inverse of K_s is

$$[K_s]^{-1} = \begin{bmatrix} \frac{1}{K} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{K} \end{bmatrix} \quad . \qquad (18)$$

Eliminate u_k between Equations 16 and 17 to get

$$\{u_g\} = [R_i]\{a\} + ([A_{ij}] + [K_s]^{-1})\{P\} \quad . \qquad (19)$$

Thus, all that is required to accommodate springs is to add the spring flexibilities to the diagonal of the spline influence coefficient matrix. This is obvious by physical reasoning, since the spring and spline flexibilities are in series and can be added directly.

17.3.4 Rigid Arms on Linear Splines

The linear splines used for geometry interpolation have rigid arms (see Figure 1). Mathematically, these represent equations of constraint between the displacements and rotations at the spline end and attachment end of the spline. The constraint equations are used to transform the influence functions from the spline ends to influence functions at the attachment ends. The complete transformed influence functions are shown in Table 1.

17.3.5 Coordinate Systems and Constraints

The spline constraints are derived in spline coordinates, transformed to the global coordinate system, then all splines are appended to a common constraint matrix, G_{kg}. Finally, the multi, single and omit constraints are applied to reduce this to G_{ka}.

9l6o

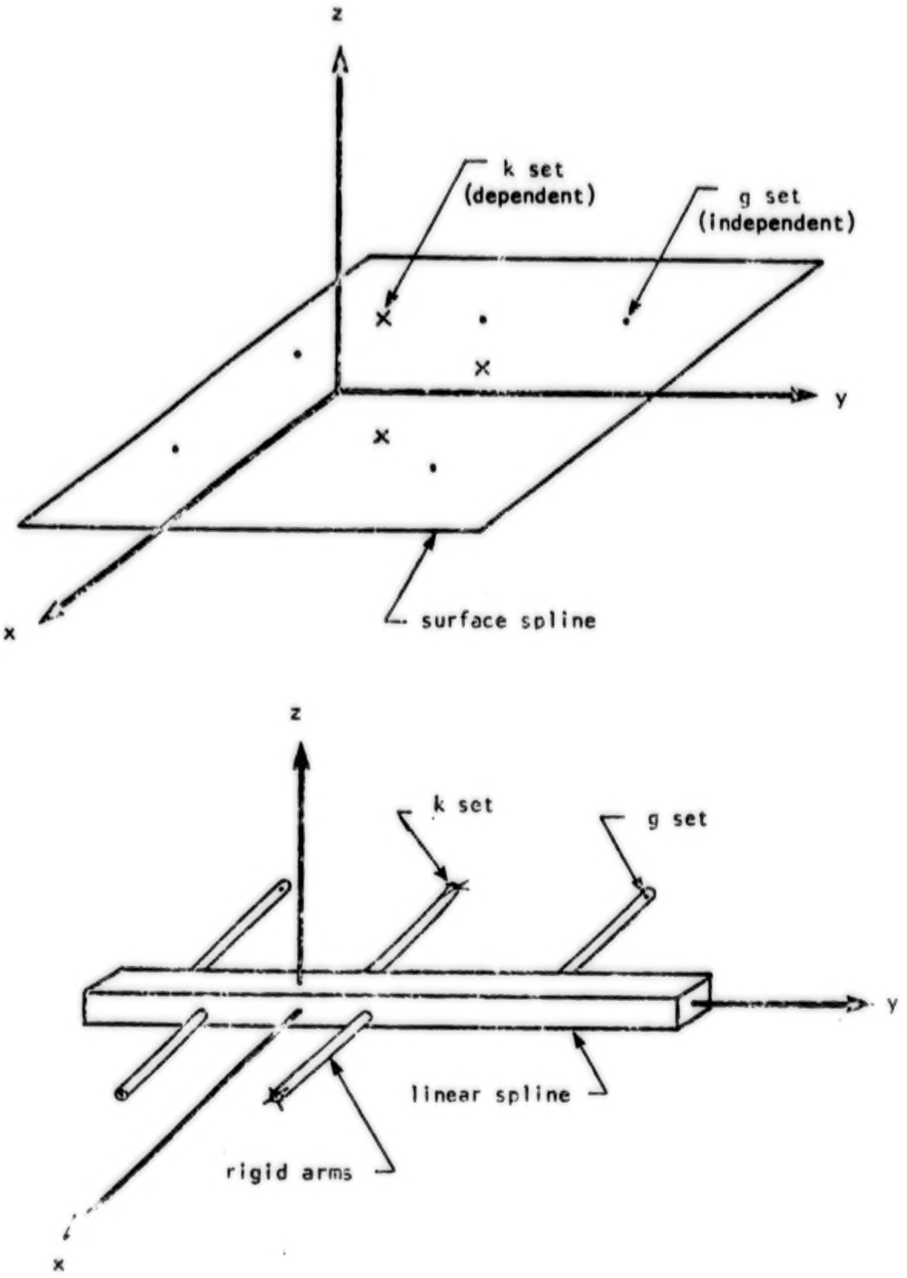

z
k set
(dependent)
g set
(independent)
y
x
surface spline
z
k set
g set
y
linear spline
rigid arms
x

Figure 1. Splines and their coordinate systems.

Table 1. Matrices for spline interpolation

The A Matrix for Surface Splines:

$$\left\{\begin{matrix} u_z \\ \theta_x \\ \theta_y \end{matrix}\right\}_i = \begin{bmatrix} \dfrac{r_{ij}^2 \ln r_{ij}^2}{16\pi D} + \dfrac{\delta_{ij}}{k_z} \\ \dfrac{(y_i-y_j)(1+\ln r_{ij}^2)}{8\pi D} \\ -\dfrac{(x_i-x_j)(1+\ln r_{ij}^2)}{8\pi D} \end{bmatrix} \{P\}_j$$

where $r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$

$$\delta_{ij} = \begin{cases} 1 \text{ for } i = j \\ 0 \text{ for } i \neq j \end{cases}$$

The R Matrix for Surface and Linear Splines

$$\left\{\begin{matrix} u_z \\ \hline \theta_x \\ \hline \theta_y \end{matrix}\right\}_i = \left[\begin{array}{c|c|c} 1 & y_i & -x_i \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right] \left\{\begin{matrix} u_z \\ \theta_x \\ \theta_y \end{matrix}\right\}_r$$

The A Matrix for Linear Splines

$$\left\{\begin{matrix} u_z \\ \hline \theta_x \\ \hline \theta_y \end{matrix}\right\}_i = \left[\begin{array}{c|c|c} \dfrac{|y_i-y_j|^3}{12EI} - \dfrac{x_i x_j |y_i-y_j|}{2GJ} + \dfrac{\delta_{ij}}{k_z} & -\dfrac{|y_i-y_j|(y_i-y_j)}{4EI} & \dfrac{x_i|y_i-y_j|}{2GJ} \\ \hline \dfrac{|y_i-y_j|(y_i-y_j)}{4EI} & -\dfrac{|y_i-y_j|}{2EI} + \dfrac{\delta_{ij}}{k_{\theta x}} & 0 \\ \hline \dfrac{x_j|y_i-y_j|}{2GJ} & 0 & -\dfrac{|y_i-y_j|}{2GJ} + \dfrac{\delta_{ij}}{k_{\theta y}} \end{array}\right] \left\{\begin{matrix} P_z \\ \hline M_x \\ \hline M_y \end{matrix}\right\}_j$$

REFERENCE

1. Harder, R. L., and Desmarais, R. N., "Interpolation Using Surface Splines," J. of Aircraft, Vol. 9, No. 2, February 1972, pp. 189-191.

17.4 FLUTTER SOLUTION TECHNIQUES

Flutter analysis is an unsteady stability problem. It may be solved for subsonic or supersonic flows by merely substituting the correct aerodynamic formulation. For the linear case, the solution involves a series of complex eigenvalue solutions.

17.4.1 The K-method of Flutter Solution

The basic equation for modal flutter analysis by the K-method is

$$[-M_{hh}\omega^2 + iB_{hh}\omega + (1+ig)K_{hh} - (\tfrac{1}{2}\rho V^2)Q_{hh}(k,m)]\{u_h\} = 0 \quad , \qquad (1)$$

where M_{hh} = modal mass matrix, usually (but not necessarily) diagonal,

B_{hh} = modal damping matrix,

K_{hh} = modal spring matrix, usually (but not necessarily) diagonal. May be complex (for real structural damping). May be singular if there are rigid body modes,

$Q_{hh}(k,m)$ = aerodynamic force matrix, which is a function of parameters k, reduced frequency, and m, Mach number,

ω = circular frequency = $2\pi f$,

g = artificial structural damping,

ρ = density,

V = velocity,

and u_h = modal amplitude vector.

The above parameters are not independent, since $k = \bar{c}\omega/2V$, where $\bar{c}$ is a reference length.

For the K-method of solution, the aerodynamic terms are converted to aerodynamic mass

$$\left[-\left[M_{hh} + \frac{\rho}{2}\left(\frac{\bar{c}}{2k}\right)^2 Q_{hh}(k,m)\right]\frac{\omega^2}{1+ig} + B_{hh}\frac{i\omega}{\sqrt{1+ig}} + K_{hh}\right]\{u_h\} = 0 \quad . \qquad (2)$$

The term involving B in Equation 2 was multiplied by $\sqrt{1+ig}$, which is valid only at flutter, i.e., when g = 0. Equation 2 is solved as an eigenvalue problem for a series of values for parameters k, m and ρ. The complex eigenvalue is $\omega^2/(1+ig)$, which can be interpreted as real values of ω and g. Velocity, V, is recovered from $V = \bar{c}\omega/2k$. Flutter occurs for values of k, m and ρ for which g = 0. The solutions are not valid except when g = 0, since the aerodynamic force terms are valid only for sinusoidal motion.

A slight variation has been used in NASTRAN. The equation is written as

$$\left[\left[\left(\frac{2k}{\bar{c}}\right)^2 M_{hh} + \left(\frac{\rho}{2}\right) Q_{hh}(k,m)\right]\left(\frac{-V^2}{1+ig}\right) + \left(\frac{2k}{\bar{c}}\right) B_{hh} \frac{iV}{\sqrt{1+ig}} + K_{hh}\right]\{u_h\} \doteq 0 \quad . \tag{3}$$

Thus, the eigenvalue is

$$p^2 = -V^2/(1+ig) \tag{4}$$

Equation 4 is approximated by

$$p = V\left(\frac{g}{2} + i\right) \quad . \tag{5}$$

The advantage of this method is that it is possible to let k = 0, since division by zero is not required. This may actually allow solution of static divergence problems.

To solve Equation 3, it must first be reduced to the canonical form for the Hessenberg method,

$$[A - \lambda I]u = 0 \quad . \tag{6}$$

To do this, use the method of Section 10.5.1.

As the flow chart for modal flutter analysis (Figure 1, Section 17.6) shows, the K-method of flutter analysis is a looping procedure. The values of V, g and f are solved for various values of k, m and ρ. Plots of V versus g can be used to determine flutter (when g goes through zero to positive values).

17.4.2 The KE-method of Flutter Solution

A more efficient K-method of flutter analysis (see Section 17.4.1) is possible if the analyst is willing to assume no viscous damping from any source, e.g., from the structure or a control system, and to restrict his solution to eigenvalues and not require eigenvectors. Then many of the operations can be done in-core with a consequent increase in efficiency. This efficient K-method algorithm is called the KE-method. With this increase in efficiency, a greater number of points on a flutter stability curve can be obtained for a given cost, and cases with poorly behaved stability curves can be studied more thoroughly. This method gives results similar to those of Desmarais and Bennett (Reference 1).

In order to sort the roots so that curves can be drawn, the roots must be ordered. For the first value of k, the roots are accepted in the order output by the eigenvalue subroutine ALLMAT.

(see Section 10.5). If the i-th eigenvalue for the n-th reduced frequency k_n is denoted by $p_{i,n}$, then an extrapolated eigenvalue may be defined as

$$p^{(e)}_{i,n} = p_{i,n-1} + (k_n - k_{n-1})(p_{i,n-1} - p_{i,n-2})/(k_{n-1} - k_{n-2}) \qquad n = 2, 3, 4, \ldots \quad (7)$$

in which $p_{i,0}$ is chosen equal to $p_{i,1}$. Then the $p_{i,n}$ are ordered according to closeness to $p^{(e)}_{i,n}$ where the "closeness" is measured by a minimum value of

$$[\mathrm{Re}(p^{(e)}_{i,n} - p_{i,n})]^2 + [\mathrm{Im}(p^{(e)}_{i,n} - p_{i,n})]^2 \qquad (8)$$

17.4.3 The PK-method of Flutter Solution

The fundamental equation for modal flutter analysis by the PK-method is

$$[M_{hh}p^2 + (B_{hh} - \tfrac{1}{4}\rho\bar{c}VQ^I_{hh}/k)p + (K_{hh} - \tfrac{1}{2}\rho V^2 Q^R_{hh})]\{u_h\} = 0 \quad , \qquad (9)$$

where M_{hh} = modal mass matrix, usually (but not necessarily) diagonal for inertial terms. Inclusion of control systems (as extra points) can introduce off-diagonal terms,

B_{hh} = modal damping matrix. An equivalent viscous structural damping matrix is usually diagonal. Control systems can introduce off-diagonal terms,

K_{hh} = modal structural stiffness matrix, usually (but not necessarily) diagonal. May be singular if there are rigid body modes. Control systems can add off-diagonal terms,

Q^I_{hh} = modal aerodynamic damping matrix, a function of reduced frequency k and Mach number m,

Q^R_{hh} = modal aerodynamic stiffness matrix, a function of reduced frequency k and Mach number m,

p = eigenvalue = $\omega(\gamma \pm i)$,

ω = circular frequency,

γ = transient decay rate coefficient,

ρ = density,

V = velocity,

$\bar{c}$ = reference chord,

k = reduced frequency = $\omega\bar{c}/2V$,

and u_h = modal amplitude vector.

The matrix terms in Equation 9 are all real. Q^R_{hh} and Q^I_{hh} are the real and imaginary parts of $Q_{hh}(k,m)$, respectively. Note that the circular frequency and the reduced frequency are not independent since $k = \omega\bar{c}/2V$.

For the PK-method of solution, Equation 9 is rewritten in a canonical form with twice the order.

$$[A - pI]\{\bar{u}_h\} = 0 \quad , \tag{10}$$

where [A] is the real matrix

$$[A] = \left[\begin{array}{c|c} 0 & I \\ \hline -M_{hh}^{-1}(K_{hh} - \tfrac{1}{2}\rho V^2 Q_{hh}^R) & -M_{hh}^{-1}(B_{hh} - \tfrac{1}{4}\rho\bar{c}VQ_{hh}^I/k) \end{array}\right] \quad , \tag{11}$$

and $\{\bar{u}_h\}$ now includes both modal displacements and velocities.

The solution of Equation 10 requires iteration so that the condition

$$k = (\bar{c}/2V)\mathrm{Im}(p) \tag{12}$$

may be satisfied. The iteration is diagrammed in Figure 1. It begins with $k = 0$ (actually a small value so that Q_{hh}^I/k can be computed) and all real roots immediately satisfy Equation 4 but the complex roots do not. For the real roots, the decay rate coefficient is

$$\gamma = p\bar{c}/(\ell n2)V \quad . \tag{13}$$

The iteration for the complex roots proceeds as follows. Let the complex pairs of eigenvalues be written as

$$p_{rs}^{(j)} = \omega_{rs}^{(j)}(\gamma_{rs}^{(j)} \pm i) \quad , \tag{14}$$

where r denotes the oscillatory mode number ordered by frequency ($\omega_{1s} < \omega_{2s} < \ldots$), s denotes the number of the oscillatory mode under investigation, and j denotes the iteration (eigenvalue solution) number, so that the next estimate of the (nonzero) reduced frequency is

$$k_s^{(j)} = \omega_{ss}^{(j)}\bar{c}/2V \quad . \tag{15}$$

The estimate of the first nonzero reduced frequency is taken as

$$k_1^{(1)} = \omega_{11}^{(1)}\bar{c}/2V \quad . \tag{16}$$

Convergence to the first oscillatory root occurs when

$$|k_1^{(j)} - k_1^{(j-1)}| < \varepsilon \quad . \tag{17}$$

Let the converged complex eigenvalues be

$$p_{rs}^{(c)} = \omega_{rs}^{(c)}(\gamma_{rs}^{(c)} \pm i) \quad , \tag{18}$$

where only p_{ss} satisfies both Equations 10 and 12. The search for the next oscillatory mode begins by increasing s by one. The first estimate of the next reduced frequency is

$$k_s^{(0)} = \omega_{s,s-1}^{(c)}\bar{c}/2V \quad , \tag{19}$$

and the iteration of Equations 14 and 16 is continued until Equation 17 is satisfied. Equations 18 and 19 begin the search for each of the higher modes of interest.

The eigenvalues and eigenvectors of Equation 10 are found by a real version of the Upper Hessenberg method described below. The order of calculations for different values of altitude density, velocity and/or Mach number is indicated in Figure 2.

The principal advantage of the PK-method over the K-method (or the KE-method) is that it produces results directly for given values of Mach number (or speed), whereas the other methods require iteration. In addition, the damping given by $\gamma_{rs}^{(c)}$ in Equation 18 is a better approximation to the decay rate than the parameter g in Equation 5.

17.4.3.1 Eigenvalue and Eigenvector Extraction for the PK-method

The Upper Hessenberg method is used to extract the real or complex conjugate eigenvalues of the real matrix in the canonical form of Equation 10. The Upper Hessenberg method is described in Section 10.5 for complex matrices. The basic reference is Wilkinson (Reference 2) for both the real and complex cases, but Parlett (Reference 3) has given a concise discussion of the algorithms for the real case which have been automated in the subroutines HSBG (Reference 4) and ATEIG (Reference 5) and modified for NASTRAN. The subroutine ATEIG utilizes the elegant procedure of Francis (Reference 6) for determining complex conjugate eigenvalues of general unsymmetric real matrices. This is known as the Double QR-Transformation and it retains entirely real arithmetic operations. The algorithm permits a double shift (see Section 10.5.5 on Shifting) to accelerate convergence after each iteration and the unitary transformation is that of Householder (Reference 2, page 533) rather than that of Givens. The algorithm also deflates (see Section 10.5.6 on Deflation) two eigenvalues at a time.

When requested, the eigenvectors are found in two iterations by the Inverse Power method with shifts (see pages 323 and 626-628, Reference 2 (see previous page, 17.4-5)) using the subroutine EGNVCT written by Dr. A. M. Cunningham, Jr., of General Dynamics, Fort Worth, Texas. The subroutine EGNVCT finds the eigenvector $\{u\}$ from the complex matrix $[A]$ and the complex eigenvalue λ by solving the equation

$$[A - \lambda I]\{u\} = 0 \quad . \tag{20}$$

NASTRAN utilizes the subroutine by setting

$$\lambda = 0 \quad , \tag{21}$$

and

$$[A] = [Mp^2 + Bp + K] \tag{22}$$

in Equation 20.

REFERENCES

1. Desmarais, R. N., and Bennett, R. M., "An Automated Procedure for Computing Flutter Eigenvalues," J. Aircraft, Vol. 11, No. 2, February 1974, pp. 75-80.

2. Wilkinson, J. H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

3. Ralston, A., and Wilf, H. S. (Ed.), Mathematical Methods for Digital Computers, Vol. II, John Wiley & Sons, Inc., New York, 1968, pp. 116-130.

4. Anon., "Subroutine HSBG," IBM 360 Library Subroutine Manual, Form H20-0205-3, 14 Feb. 1969.

5. Anon., "Subroutine ATEIG," IBM 360 Library Subroutine Manual, Form H20-0205-3, 14 Feb. 1969.

6. Francis, J. G. F., "The QR Transformation - a Unitary Analogue to the LR Transformation," Parts 1 and 2, Computer Journal, Vol. 4, No. 3 (October 1961), pp. 265-271, and No. 4 (January 1962), pp. 332-345.

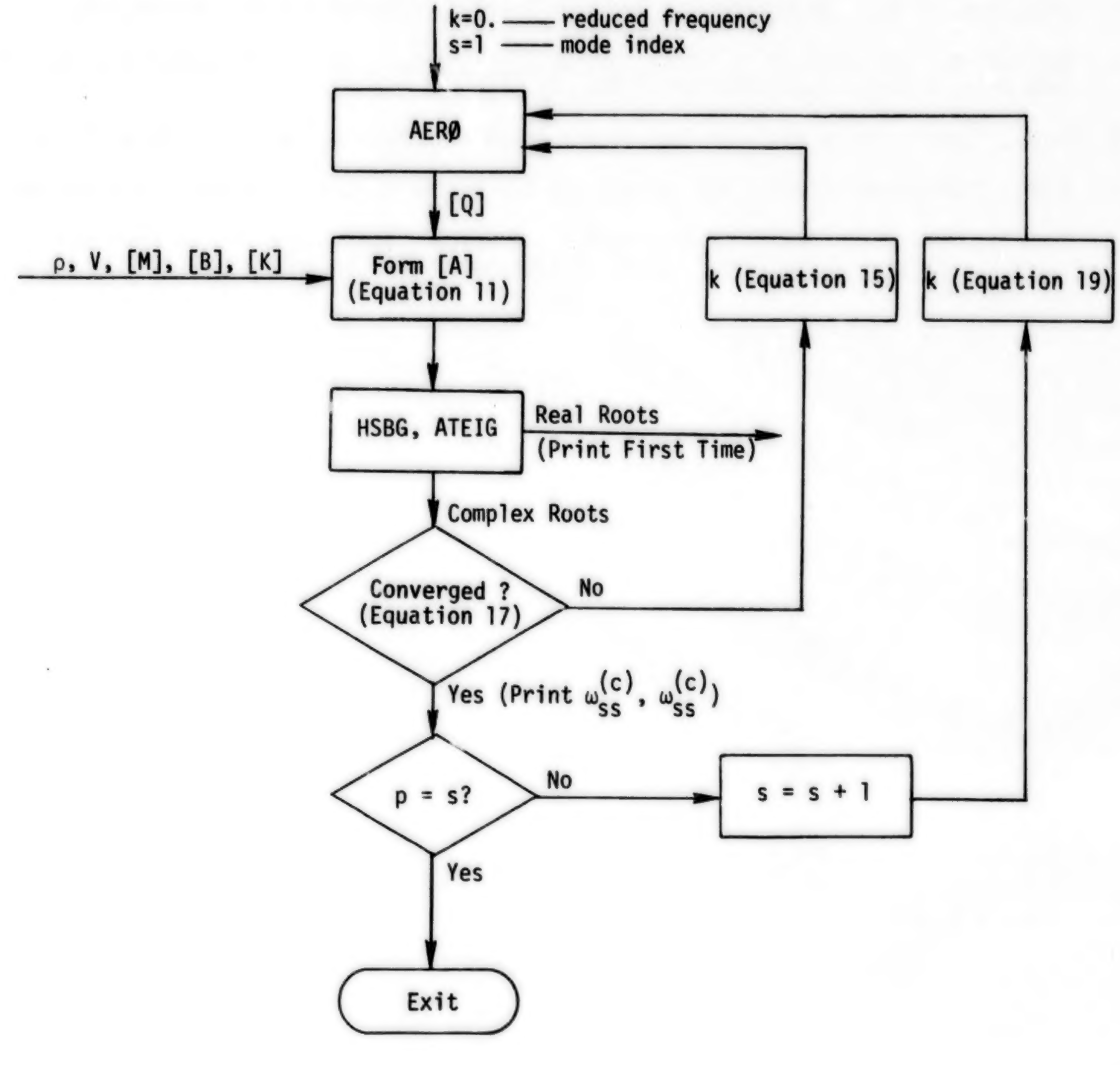

k=0. —— reduced frequency
s=1 —— mode index
AERØ
[Q]
ρ, V, [M], [B], [K]
Form [A]
(Equation 11)
k (Equation 15)
k (Equation 19)
HSBG, ATEIG
Real Roots
(Print First Time)
Complex Roots
Converged ?
(Equation 17)
No
Yes (Print $\omega_{ss}^{(c)}$, $\omega_{ss}^{(c)}$)
p = s?
No
s = s + 1
Yes
Exit

Figure 1. Basic flow diagram for PK-Flutter method.

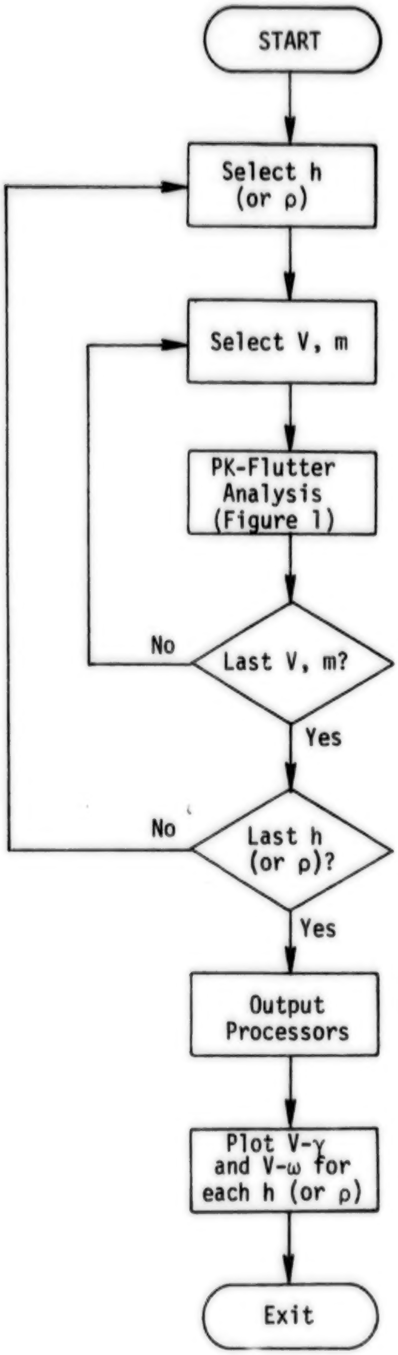

START
Select h
(or ρ)
Select V, m
PK-Flutter
Analysis
(Figure 1)
No
Last V, m?
Yes
No
Last h
(or ρ)?
Yes
Output
Processors
Plot V-γ
and V-ω for
each h (or ρ)
Exit

Figure 2. Complete flow diagram for PK-Flutter analysis.

17.5 UNSTEADY AERODYNAMIC THEORIES

Three matrix equations summarize the relationships required to define a set of aerodynamic influence coefficients (Reference 1). These are the basic relationships between pressure and downwash (normal-wash),

$$\{w_j\} = [A_{jj}]\{P_j\} \quad , \tag{1}$$

the substantial differentiation of the deflections to obtain downwash,

$$\{w_j\} = [D^1_{jk} + ik\, D^2_{jk}]\{u_k\} \quad , \tag{2}$$

and the integration of the pressure to obtain forces,

$$\{P_k\} = [S_{kj}]\{P_j\} \quad , \tag{3}$$

where u_k, P_k = displacement and forces at aerodynamic grid points,
w_j = downwash (normalwash),
P_j = pressure on lifting element,
$A_{jj}(k,m)$ = aerodynamic influence matrix,
D^1_{jk}, D^2_{jk} = substantial differentiation matrices,
and S_{kj} = integration matrix.

In general, the matrices A, D^1, D^2, and S are computed, although for some aerodynamic theories A^{-1} is obtained directly; in the case of extra points, e.g., when an active control system is included, some partitions of the matrices D^1 and D^2 must be supplied by the user. The matrix A (or A^{-1}) is computed for a user supplied list of (k,m).

17.5.1 Doublet-Lattice Method

The Doublet-Lattice Method (DLM) can be used for interfering lifting surfaces in subsonic flow. The theory is presented thoroughly in the literature (References 2, 3, and 4) and will not be reproduced here. The following general remarks summarize the essential features of the method.

PRECEDING PAGE BLANK NOT FILMED

PAGE 979 INTENTIONALLY BLANK

The theoretical basis of the DLM is linearized aerodynamic theory. The undisturbed flow, about which the perturbations occur, is uniform, i.e., no spatial variations, and either steady or gusting harmonically. Thus, all lifting surfaces must lie nearly parallel to the flow. The DLM is an extension of the steady Vortex-Lattice Method to unsteady flow.

Each of the interfering surfaces is divided into small trapezoidal lifting elements ("boxes") such that the boxes are arranged in strips parallel to the free stream so that surface edges, fold lines, and hinge lines lie on box boundaries. The unknown lifting pressures are assumed to be concentrated uniformly across the one-quarter chord line of each box. There is one control point per box and the surface normalwash boundary condition is satisfied at each of these points. The control point is centered spanwise on the three-quarter chord line of the box.

The code for computing A_{jj} was taken directly from Giesing, Kálmán, and Rodden (Reference 5). Any number of arbitrarily shaped interfering surfaces can be analyzed, provided that each is idealized as one or more trapezoidal planes. Various symmetry options are also available. These include geometry (planform), motion (symmetric or antisymmetric), and ground-effect.

17.5.2 Subsonic Wing-Body Interference Theory

The method of images, along with Slender Body Theory has been added to the Doublet-Lattice Method (DLM) by Giesing, Kálmán, and Rodden (References 5, 6, and 7). The DLM is used to represent the configuration of interfering lifting surfaces, while Slender Body Theory is used to represent the lifting characteristics of each body, i.e., fuselage, nacelle, or external store. The primary wing-body interference is accounted for by a system of images of the DLM vortices and doublets within a cylindrical interference body that circumscribes each slender body. The secondary wing-body interference that results from the DLM bound vortices and doublets is accounted for by a line of doublets located on the longitudinal axis of each slender body. The boundary conditions of no flow through the lifting surfaces or through the body (on the average about the periphery) leads to the equations for the lifting pressures on the surfaces and for the longitudinal (and/or lateral) loading on the bodies in terms of the normalwashes on the wing-body combination.

The code for computing the aerodynamic matrices was adapted for NASTRAN from Reference 4. The adaptation required matrix formulation of all of the body interference and body loading calculations. These equations are written using the symbols adopted for NASTRAN and showing the equivalences to names used in the documentation of Reference 4.

The program of Reference 4 finds the forces on the lifting boxes and bodies of an idealized airplane in terms of the motions of these elements. The lifting surfaces are divided into boxes. The bodies are divided into elements. There are two types of body elements: slender elements are used to simulate a body's own motion; and interference elements are used to simulate the interaction with other bodies and boxes. The body elements may have y (lateral), z (vertical), or both degrees of freedom.

The basic method is the superposition of singularities and their images. There are two basic types, which are "forces" and modified acceleration potential "doublets." Each "force" singularity is shown to be equivalent to a line of doublets in the wake. The wing boxes use the "force" type of singularity concentrated along the box quarter chord. The interference elements use the "doublet" type of singularity. The slender elements use both types. The first equation relates the downwashes to the singularities:

$$\left\{\begin{array}{c} w_w \\ \hline 0 \\ \hline w_s \end{array}\right\} = \left[\begin{array}{c|c|c} A_{ww} & A_{wI} & A_{ws} \\ \hline A_{Iw} & A_{II} & A_{Is} \\ \hline 0 & 0 & A_{ss} \end{array}\right] \left\{\begin{array}{c} C_w \\ \hline \mu_I \\ \hline \mu_s \end{array}\right\} , \tag{4}$$

where w_w = the wing box downwashes at the 3/4-chord,

0 = the downwashes for interference elements due to singularities other than slender elements in the same body,

w_s = the downwashes for slender body elements,

C_w = the forces distributed along wing box quarter chords,

μ_I = the interference doublets,

μ_s = the slender element doublets,

$\left[\begin{array}{c|c} A_{ww} & A_{wI} \\ \hline A_{Iw} & A_{II} \end{array}\right]$ = [DT], Section 5.3.1, Reference 4,

$$\left[\begin{array}{c} A_{ws} \\ \hdashline A_{Is} \end{array}\right] = [DZ \,\vdots\, DY]\text{, Section 5.5.3, Reference 4,}$$

and $[A_{ss}] = \left\lceil D2D^{-1} \right\rfloor$, which is not matricized but is diagonal and will be discussed later.

The next equation relates the forces to the singularities:

$$\left\{\begin{array}{c} f_w \\ \hdashline f_s \end{array}\right\} = \left[\begin{array}{c:c:c} S_{ww} & 0 & 0 \\ \hdashline S_{sw} & S_{sI} & S_{ss} \end{array}\right] \left\{\begin{array}{c} C_w \\ \hdashline \mu_I \\ \hdashline C_s \end{array}\right\} \quad , \qquad (5)$$

where C_s = the slender element forces,

$\left\lceil S_{ww} \right\rfloor$ = box areas (not matricized, but diagonal), and

$[S_{sw} \,\vdots\, S_{sI} \,\vdots\, S_{ss}]$ = [BFS], Section 5.8.1, Reference 4, but rearranged in the order of the rows. These equations use Method 1 of Reference 4; Method 2 cannot be used since it is not matricized. All of the above matrices have been modified to include the images of the sources on the symmetry plane. Reference 4 uses the convention that images are not used for bodies or surfaces on a plane of symmetry.

In the slender body part of the program developed in Reference 4, there is no matrix which relates the slender body forces, C_s, to the slender body modified doublets, μ_s. This relationship involves only elements of the same body and of the same orientation. Hence, we must derive this matrix. The differential equation relating distributions of these is

$$C(x) = \left(\frac{\partial}{\partial x} + i\,\frac{\omega}{U}\right)\mu(x) \quad , \qquad (6)$$

where $C(x)$ = lift per unit length/dynamic pressure (i.e., has units of length),

$\mu(x)$ = doublet strength per unit length/free stream velocity (i.e., has units of length^2),

x = axial coordinates,

and ω/U = reduced frequency.

The elements of the vector $\{C_s\}$ are $\Delta x \cdot C(x_{center})$, which are the total forces on the elements divided by the dynamic pressure. The elements of $\{\mu_s\}$ are $\mu(x_{center})$. In subroutine MUZYC (Section 5.5.4, Reference 4), the values of C_s (called ΔCpΔA in the reference) are evaluated from an equation which is equivalent to

$$C_s = \Delta x\left[\left(\frac{d}{dx} + i\,\frac{\omega}{U}\right)\mu(x)\right]_{x=x_{center}} \quad , \tag{7}$$

$$\mu(x) = (D2D)^{-1}\, w_s(x) \quad , \tag{8}$$

and

$$w_s(x) = \left(\frac{d}{dx} + i\,\frac{\omega}{U}\right)\Sigma A_n(x_s/\bar{c})^n \quad , \tag{9}$$

where $(D2D)^{-1} = 2\pi(1+AR)a_o^2(x)$ (z bodies),

w_s = downwash (dimensionless),

$a_o(x)$ = halfwidth (radius),

AR = height/width ratio of body,

and $\Sigma A_n(x_s/\bar{c})^n$ = polynomial mode shape, which is input data in Reference 4.

The method used was to derive a matrix, [G], which relates the vector of $\{C_s\}$ to $\{\mu_s\}$,

$$\{C_s\} = [G]\{\mu_s\} \quad . \tag{10}$$

The derivation of [G] assumes that μ/a_o^2, which is proportional to w, is a smooth function. Thus, $\mu(x) = a_o^2(x)\cdot[\mu(x)/a_o^2(x)]$. Using Equation 6,

$$C(x) = \left(\frac{\mu}{a_o^2}\right)\frac{\partial}{\partial x}\,(a_o^2) + (a_o^2)\left(\frac{\partial}{\partial x} + i\,\frac{\omega}{U}\right)\left(\frac{\mu}{a_o^2}\right)$$

$$= \left(\frac{2}{a_o}\,\frac{da_o}{dx} + i\,\frac{\omega}{U}\right)\mu + a_o^2\,\frac{\partial}{\partial x}\left(\frac{\mu}{a_o^2}\right) \quad . \tag{11}$$

The numerical derivatives required for the last term in Equation 11 are evaluated by the following rules:

One point per body: derivative = 0

Two or more points per body: derivative = $\begin{cases}\text{2 point rule at end points}\\ \text{3 point rule at interior points}\end{cases}$

The 2 point rule comes from a linear fit, the three point rule from a quadratic fit. Examples are,

$$\left(\frac{dy}{dx}\right)_1 = \frac{y_2 - y_1}{x_2 - x_1} \qquad \text{(2-point)} \quad , \tag{12}$$

and

$$\left(\frac{dy}{dx}\right)_2 = \frac{y_2 - y_1}{x_2 - x_1} + \frac{y_3 - y_2}{x_3 - x_2} - \frac{y_3 - y_1}{x_3 - x_1} \qquad \text{(3-point)} \quad . \tag{13}$$

Using this, the part of [G] for one body is given by

$$[G_{ij}] = (\Delta x)_i \left[\delta_{ij} \left(\frac{2}{a_0} \frac{da_0}{dx} + i \, \frac{2k}{\bar{c}} \right)_i + g_{ij} \right] \quad , \tag{14}$$

where g_{ij} is tridiagonal and is given by

$$g_{ij} = \frac{(a_0^2)_i}{(a_0^2)_j} \begin{cases} \dfrac{1}{x_{i+1} - x_{i-1}} - \dfrac{1}{x_i - x_{i-1}} & j = i-1 \\ \dfrac{1}{x_i - x_{i-1}} - \dfrac{1}{x_{i+1} - x_i} & j = i \\ \dfrac{1}{x_{i+1} - x_i} - \dfrac{1}{x_{i+1} - x_{i-1}} & j = i+1 \\ 0 & \text{otherwise} \end{cases} \quad , \tag{15}$$

For the first element (i=1), delete terms involving (i-1). For the last point (i=N), delete terms involving x_{i+1}.

Equation 10 is used to eliminate the slender element forces C_s from Equation 5, giving

$$\begin{Bmatrix} f_w \\ \hline f_s \end{Bmatrix} = \left[\begin{array}{c|c|c} S_{ww} & 0 & 0 \\ \hline S_{sw} & S_{sI} & S_{ss}G \end{array} \right] \begin{Bmatrix} C_w \\ \hline \bar{\mu}_I \\ \hline \bar{\mu}_s \end{Bmatrix} = [S_{kj}]\{P_j\} \quad . \tag{16}$$

Equations 4 and 16 relate forces to downwashes. The relationship of deflections to downwashes is given by Equation 2. As can be seen from Equation 4, there is zero "downwash" for all interference body elements, hence the rows of D^1_{jk} and D^2_{jk} associated with interference body elements vanish. All other rows represent total derivatives for downwashes of boxes and slender body elements. The basic form of Equations 1 through 3 is kept, even in the case of panels with interference and slender bodies.

17.5.3 Mach Box Method

The Mach Box Method (MBM) can be used to estimate generalized aerodynamic forces on isolated planar wings with two (adjacent) control surfaces oscillating in supersonic flow. The MBM is based on the box method first proposed by Pines, Dugundji, and Neuringer (Reference 8) as refined by Moore and Andrew (Reference 9) and by Donato and Huhn (Reference 10). The general features of the method are summarized in the following remarks.

The MBM is a numerical solution of the linearized, three-dimensional, oscillatory, supersonic perturbation potential flow equation. The regions disturbed by the lifting surface are divided into a grid of rectangular lifting elements, called Mach boxes, i.e., rectangles whose diagonals are parallel to the Mach lines. The regions divided into Mach boxes include the wing and control surfaces, and regions off the lifting surface affected by any subsonic edge of the surface, i.e., an edge whose normal Mach number is less than unity. Calculations are made of the influence of unit sources distributed uniformly over the area of each box on the aerodynamic loading at the center of every box, assuming that the constant source strength on each box is equal to the value at the box center. These calculations generate the aerodynamic influence coefficients for the surface. The coefficients obtained are velocity potential influence coefficients and are independent of the mode shapes of the structure. They are combined with the source strengths associated with a prescribed mode to obtain the corresponding velocity potential distribution over the planform. The knowledge of the modal velocity potential distributions and the mode shapes permits the calculation of the generalized aerodynamic force coefficients. The accuracy of the MBM depends on the validity of supersonic linearized theory which is generally assumed to be in the Mach number range from about 1.2 up to 3.0. At high Mach numbers, the results approach those from first order Piston Theory.

The supersonic Mach Box code used by NASTRAN is based on subroutines of a modified version of the program of Reference 10; the modifications were made by L. V. Andrew, G. V. Owens, and J. W. Sleison of the Space Division of Rockwell International. The subroutines have been adapted to calculate the aerodynamic influence coefficients A_{jj}^{-1} directly. The general planform that can be analyzed is shown in Figure 1. The following options are available to the user:

1. Leading- and/or trailing-edge cranks;
2. None, one or two adjacent trailing-edge control surfaces with swept-back hinge lines; and
3. Symmetric or antisymmetric motion (no harmonic gust fields are considered).

The freestream velocity is parallel to the x-axis as shown in Figure 1.

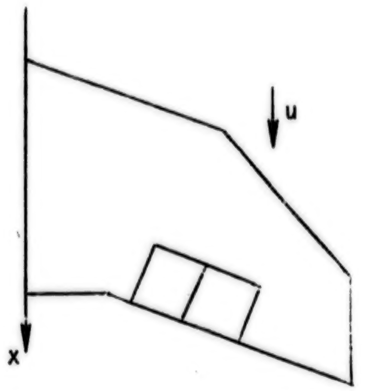


Figure 1. Planform geometry for mach box method.

17.5.4 Strip Theory

The first solutions to the unsteady theoretical aerodynamic problem were obtained in two dimensions (References 11 and 12). These solutions were utilized in flutter analysis by assuming that the loads at each spanwise station of a wing depended only on the motion of that station. Practical flutter analysis (Reference 13) divided the wing into a number of strips and the aerodynamic loads on each strip were calculated on the basis of the two-dimensional coefficients evaluated at the centerline of the strip. This "Strip Theory" was surprisingly accurate although its successes were primarily in applications to unswept wings with high aspect ratios.

The NASTRAN code for computing A_{jj}^{-1} is based on Reference 12 and is an extension of a program written by E. Albano of the Northrop Corporation. Although Reference 12 includes a trim tab, only the airfoil and an aerodynamically balanced control surface are considered; both the airfoil and the control surface are assumed to be rigid in the chordwise direction, and the control surface hinge line is assumed to remain on the wing chord line, i.e., no camber motions or hinge failures are considered. The Theodorsen function for the unsteady circulatory loads is written as

$$C(k) = F(k) + iG(k) \quad , \qquad (17)$$

where $k = \omega b/V$ is the local reduced frequency, i.e., b is the semi-chord of the strip. The user has the option of calculating F(k) and G(k) from their exact expressions in terms of Bessel

ORIGINAL PAGE IS
OF POOR QUALITY

functions, or from approximations of the form

$$F(k) = \sum_{n=0} \frac{b_n}{1 + (\beta_n/k)^2} \quad , \tag{18}$$

and

$$G(k) = \sum_{n=1} \frac{b_n(\beta_n/k)}{1 + (\beta_n/k)^2} \quad , \tag{19}$$

in which $\beta_0 = 0$. The choice of values for the parameters b_n and β_n is left to the user. In this way, Strip Theory can be adjusted to account for compressibility or aspect ratio effects. Some values of b_n and β_n are tabulated on pages 350 and 394 of Reference 14. An approximate sweep correction is also incorporated. The correction is the factor cos Λ, where Λ is the one-quarter chord line sweep angle for the aerodynamic macro element defined on a Bulk Data card, and is applied as a multiplier to all loads acting on the element.

17.5.5 Piston Theory

In the limit of high Mach number ($m^2 >> 1$) or high reduced frequency ($m^2k^2 >> 1$), the three-dimensional pressure-downwash relationship on a lifting surface becomes a nonlinear point relationship. The nonlinear point relationship can be linearized for small oscillatory disturbances while retaining the nonlinear aspects of the steady-state condition. The result is known as third order Piston Theory and has been developed by Ashley and Zartarian (Reference 15).

A computer program to obtain A_{jj}^{-1} has been written by Rodden et al. (Reference 16) in the form of a strip theory (see Section 17.5.4). This computer code has been added directly to NASTRAN with only minor adjustments for sign conventions. It has the following general features. It is an extension of Reference 15 to account for sweep and steady angle of attack and to lower the lower supersonic Mach number limit so that agreement with Van Dyke (Reference 17) is obtained through the second-order terms. A rigid chord is assumed as well as a rigid control surface hinged at its leading edge, i.e., no aerodynamic balance is considered since balance is not desirable on supersonic vehicles. Experimental correlations indicated the validity of Piston Theory in the range of Mach numbers from about 2.5 to 7.0.

REFERENCES

1. Rodden, W. P., and Revell, J. D., "The Status of Unsteady Aerodynamic Influence Coefficients," I.A.S. Fairchild Fund Paper No. FF-33, 23 January 1962.

2. Albano, E., and Rodden, W. P., "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows," AIAA J., Vol. 7, No. 2, February 1969, pp. 279-285, and Vol. 7, No. 11, November 1969, p. 2192.

3. Rodden, W. P., Giesing, J. P., and Kálmán, T. P., "Refinement of the Nonplanar Aspects of the Subsonic Doublet-Lattice Lifting Surface Method," J. Aircraft, Vol. 9, No. 1, January 1972, pp. 69-73.

4. Giesing, J. P., Kálmán, T. P., and Rodden, W. P., "Subsonic Unsteady Aerodynamics for General Configurations; Part I, Vol I - Direct Application of the Nonplanar Doublet Lattice Method," Air Force Flight Dynamics Laboratory Report No. AFFDL-TR-71-5, Part I, Vol. I, November 1971.

5. Giesing, J. P., Kálmán, T. P., and Rodden, W. P., "Subsonic Unsteady Aerodynamics for General Configurations; Part II, Vol. II - Computer Program N5KA," Air Force Flight Dynamics Laboratory Report No. AFFDL-TR-71-5, Part II, Vol. II, April 1972.

6. Giesing, J. P., Kálmán, T. P., and Rodden, W. P., "Subsonic Unsteady Aerodynamics for General Configurations; Part II, Vol. I - Application of the Doublet-Lattice Method and the Method of Images to Lifting-Surface/Body Interference," Air Force Flight Dynamics Laboratory Report No. AFFDL-TR-71-5, Part II, Vol. I, April 1972.

7. Giesing, J. P., Kálmán, T. P., and Rodden, W. P., "Subsonic Steady and Oscillatory Aerodynamics for Multiple Interfering Wings and Bodies," J. Aircraft, Vol. 9, No. 10, October 1972, pp. 693-702.

8. Pines, S., Dugundji, J., and Neuringer, J., "Aerodynamic Flutter Derivatives for a Flexible Wing with Supersonic and Subsonic Edges," J. Aero. Sci., Vol. 22, No. 10, October 1955, pp. 693-700.

9. Moore, M. T., and Andrew, L. V., "Unsteady Aerodynamics for Advanced Configurations, Part IV - Application of the Supersonic Mach Box Method to Intersecting Planar Lifting Surfaces," Air Force Flight Dynamics Laboratory Report No. FDL-TDR-64-152, Part IV, February 1965.

10. Donato, V. W., and Huhn, C. R., Jr., "Supersonic Unsteady Aerodynamics for Wings with Trailing-edge Control Surfaces and Folded Tips," Air Force Flight Dynamics Laboratory Report No. AFFDL-TR-68-30, January 1968.

11. Theodorsen, T., "General Theory of Aerodynamic Instability and the Mechanism of Flutter," NACA Report 496, 1935.

12. Küssner, H. G., and Schwarz, I., "Der schwingenden Flugel mit aerodynamisch ausgeglichenem Ruder," Luftfahrtforschung, Vol. 17, No. 11/12, Dec. 1940, pp. 337-354; translated as "The Oscillating Wing with Aerodynamically Balanced Elevator," NACA TM-991, 1941.

13. Smilg, B., and Wasserman, L. S., "Application of Three-Dimensional Flutter Theory to Aircraft Structures," Army Air Force Technical Report 4798, July 1942.

14. Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., Aeroelasticity, Addison-Wesley Publishing Co., Reading, Mass., 1955.

15. Ashley, H., and Zartarian, G., "Piston Theory - A New Aerodynamic Tool for the Aeroelastician," J. Aero. Sci., Vol. 23, No. 12, December 1956, pp. 1109-1118.

16. Rodden, W. P., Farkas, E. F., Malcom, H. A., and Kliszewski, A. M., "Aerodynamic Influence Coefficients from Piston Theory: Analytical Development and Computational Procedure," Aerospace Corp., Report No. TDR-169(3230-11)TN-2, 15 August 1962.

17. Van Dyke, M. D., "A Study of Second-Order Supersonic Flow Theory," NACA Report 1081, 1952.

17.6 AEROELASTIC RIGID FORMATS

Aeroelastic modules are arranged into two rigid formats. The flutter analysis rigid format is shown in Figure 1. The following capabilities are provided:

1. Complete modal flutter analysis. The user supplies a finite element model for structure and aerodynamics. V-g and V-f curves are drawn. If desired, flutter mode shapes may be plotted.
2. Three methods of flutter analysis are available, see Section 17.4.
3. The analysis can proceed in steps with intermediate output to check partial results. This would use the RESTART procedure and could involve rigid format switch.
4. A change of mass or spring without recalculating modes, by means of DMIG data.
5. Selection of the number of modes. A change of modes does not require aerodynamics to be recalculated.
6. The matrix values for additional Mach numbers and reduced frequencies can be appended during restart.
7. The full extra point, DMIG, transfer function capability is available for control systems, etc. Special provision is made for user supplied matrix of downwash due to motion of extra points.
8. Restart tables are provided, which allow modules to be turned on during RESTART due to changes in the Bulk Data Deck.
9. Data recovery is available for structural displacements, constraint forces and element stresses and forces for any eigensolutions selected.
10. Output of the modal amplitudes may be requested using a SDISP request in the Case Control Deck.

The aeroelastic response dynamic rigid format is shown in Figure 2. The following capabilities are provided:

1. Choice of Frequency Response, Random Analysis or Transient (by Fourier transform) Analysis is provided. Transient is defined by supplying transient type load cards and time step data. Frequency lists are required for all methods.

2. Complete RESTART/Rigid Format Switch capability.
3. Choice of loading by specified loads or gust (Doublet-Lattice Theory only).
4. Selection of Times/Frequencies for recovery of physical data.
5. Use of modal participation type of data recovery.
6. Output may be displacements, stresses, or constraint forces. Both deformed structure and XY plots are available. Aerodynamic data (pressures) are also available in frequency response.
7. Random analysis for power spectral density, rms values and zero crossings.

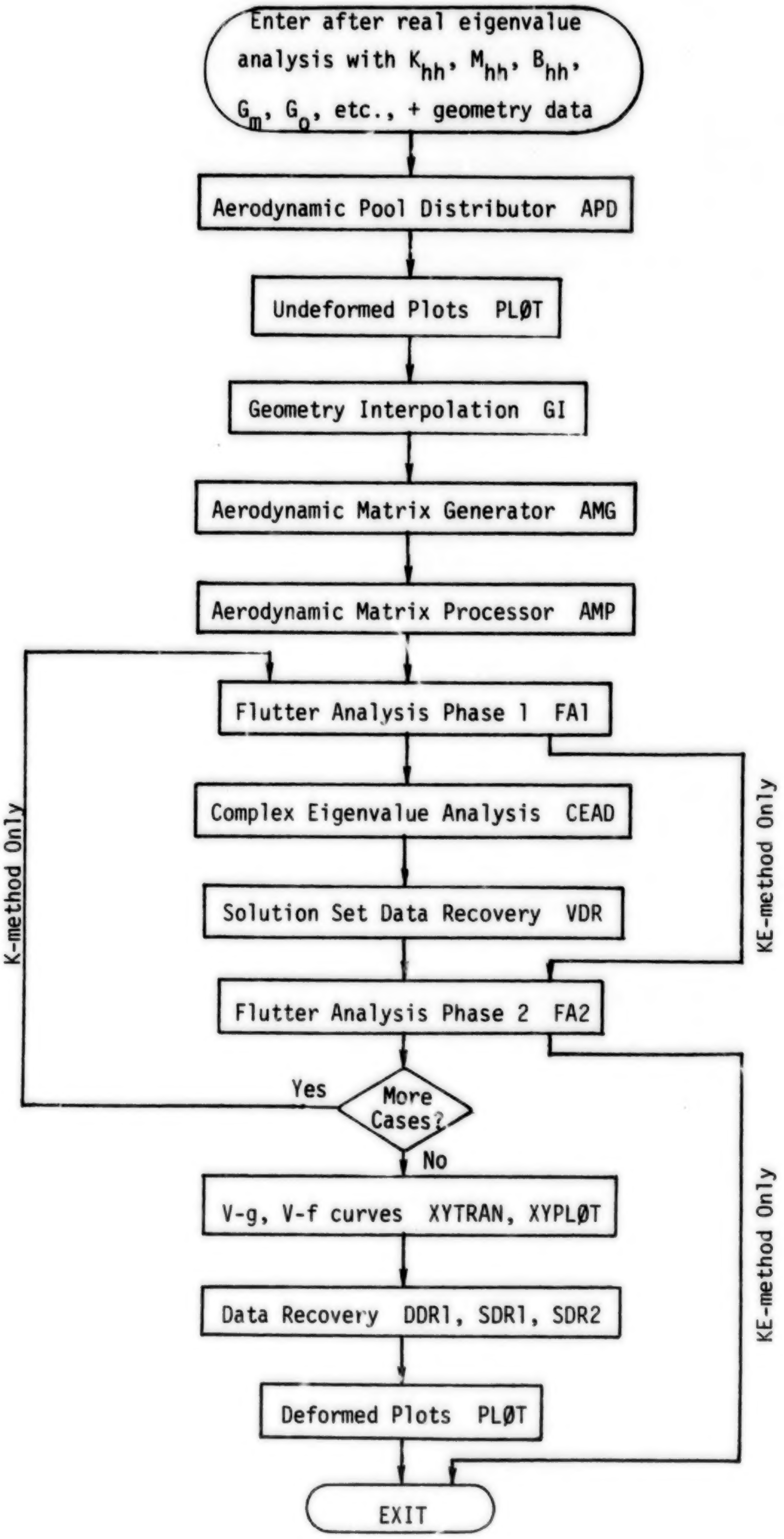

Enter after real eigenvalue analysis with K_{hh}, M_{hh}, B_{hh}, G_m, G_o, etc., + geometry data
Aerodynamic Pool Distributor APD
Undeformed Plots PLØT
Geometry Interpolation GI
Aerodynamic Matrix Generator AMG
Aerodynamic Matrix Processor AMP
Flutter Analysis Phase 1 FA1
Complex Eigenvalue Analysis CEAD
Solution Set Data Recovery VDR
Flutter Analysis Phase 2 FA2
More Cases?
Yes
No
V-g, V-f curves XYTRAN, XYPLØT
Data Recovery DDR1, SDR1, SDR2
Deformed Plots PLØT
EXIT
K-method Only
KE-method Only
KE-method Only

Figure 1. Modal flutter analysis rigid format.

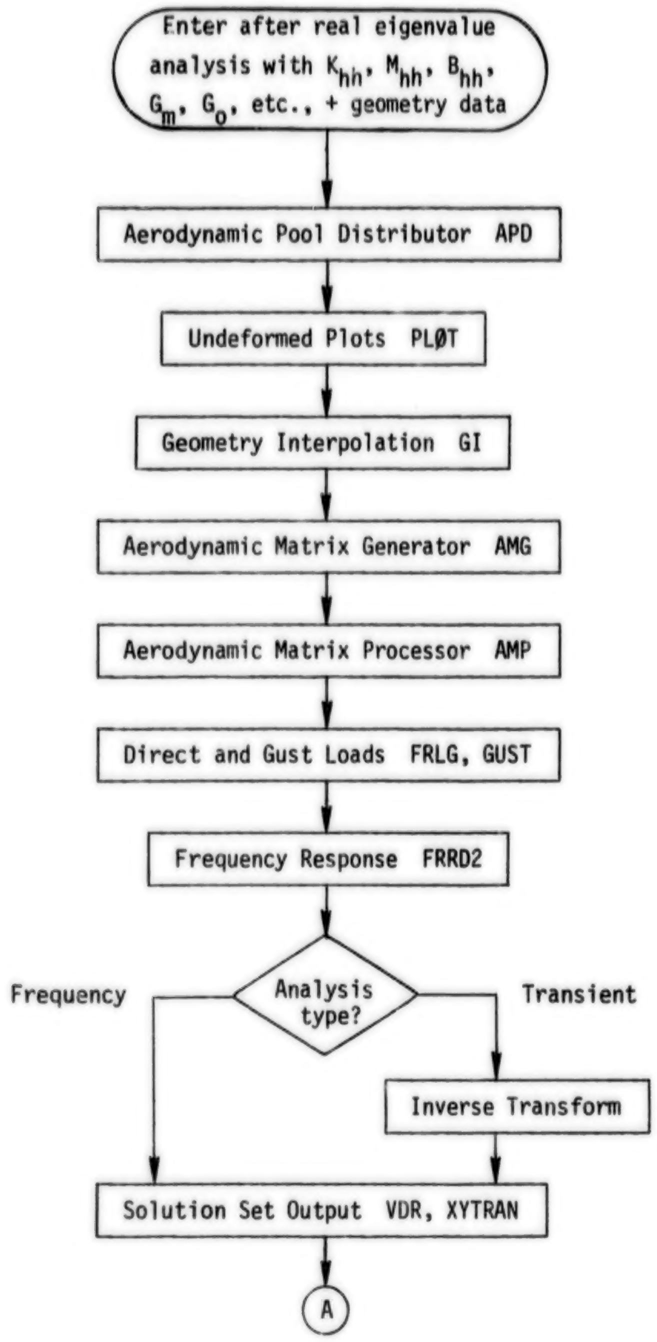

Enter after real eigenvalue analysis with K_{hh}, M_{hh}, B_{hh}, G_m, G_o, etc., + geometry data
Aerodynamic Pool Distributor APD
Undeformed Plots PLOT
Geometry Interpolation GI
Aerodynamic Matrix Generator AMG
Aerodynamic Matrix Processor AMP
Direct and Gust Loads FRLG, GUST
Frequency Response FRRD2
Analysis type?
Frequency
Transient
Inverse Transform
Solution Set Output VDR, XYTRAN
A

Figure 2. Modal aeroelastic response rigid format.

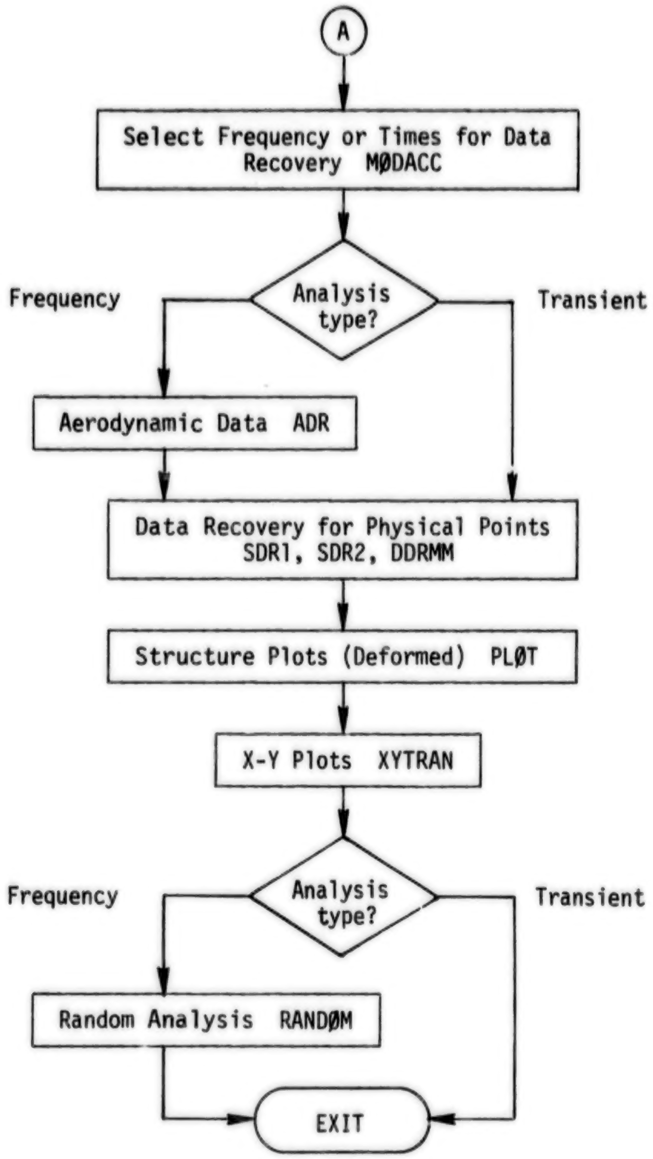

A
Select Frequency or Times for Data Recovery MODACC
Analysis type?
Frequency
Transient
Aerodynamic Data ADR
Data Recovery for Physical Points SDR1, SDR2, DDRMM
Structure Plots (Deformed) PLOT
X-Y Plots XYTRAN
Analysis type?
Frequency
Transient
Random Analysis RANDOM
EXIT

Figure 2. Modal aeroelastic response rigid format (continued).

END

DATE

FILMED

MAY 1 1984